TKK Dissertations 189
Espoo 2009

SIMULATION OF UNSTEADY FREE SURFACE FLOWS – CODE VERIFICATION AND DISCRETISATION ERROR

Doctoral Dissertation

Tommi Mikkola

Helsinki University of Technology
Faculty of Engineering and Architecture
Department of Applied Mechanics

TKK Dissertations 189
Espoo 2009

SIMULATION OF UNSTEADY FREE SURFACE FLOWS – CODE VERIFICATION AND DISCRETISATION ERROR

Doctoral Dissertation

Tommi Mikkola

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Engineering and Architecture for public examination and debate in Auditorium K216 at Helsinki University of Technology (Espoo, Finland) on the 6th of November, 2009, at 12 noon.

Helsinki University of Technology
Faculty of Engineering and Architecture
Department of Applied Mechanics

Teknillinen korkeakoulu
Insinööritieteiden ja arkkitehtuurin tiedekunta
Sovelletun mekaniikan laitos

Distribution:
Helsinki University of Technology
Faculty of Engineering and Architecture
Department of Applied Mechanics
P.O. Box 5300
FI - 02015 TKK
FINLAND
URL: http://appmech.tkk.fi/en/
Tel. +358-9-451 3501
Fax +358-9-451 4173
E-mail: leila.silonsaari@tkk.fi

ISBN 978-952-248-136-8
ISBN 978-952-248-137-5 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2009/isbn9789522481375/

TKK-DISS-2659

Yliopistopaino
Helsinki 2009

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNOLOGY P. O. BOX 1000, FI-02015 TKK http://www.tkk.fi

Author Tommi Mikkola

Name of the dissertation

Simulation of unsteady free surface flows – code verification and discretisation error

Manuscript submitted 22.06.2009	Manuscript revised 30.9.2009
Date of the defence 06.11.2009	
☒ Monograph	☐ Article dissertation (summary + original articles)
Faculty	Faculty of engineering and architecture
Department	Applied mechanics
Field of research	Marine Technology and Naval Architecture
Opponent(s)	Professor Luís Eça and Professor George Tzabiras
Supervisor	Professor Jerzy Matusiak
Instructor	Professor Jerzy Matusiak

Abstract

In this work a numerical method for the solution of unsteady, inviscid free surface flows is developed. The method is verified and the behaviour of the error related to the numerical method, as the discretisation is refined, is studied in detail. The work divides into two distinct parts. The first one focuses on the development of the solution method. The method is based on unstructured, two dimensional finite volume method. The free surface boundary conditions are satisfied on the instantaneous free surface and the computational grid tracks the deformation of the surface. Typically, in comparable methods the flow and free surface solutions are solved by time integrating the governing equations in two separate stages, which are iterated. The decoupling of the solutions limits the allowable time step in the integration, which makes the approach computationally expensive. In this work two different approaches are presented for the coupling of the solutions, which relax the time step restriction. The approaches that are proposed differ significantly from the coupling approaches presented previously in the literature in that the implementation into the existing pressure correction type solvers is straightforward. The second part concentrates on the verification of the implementation of the numerical method, i.e. on code verification, and on the investigation of the error related to the discretisation of the continuous problem. In both cases, the analysis is based on the method of manufactured solutions (MMS), in which the governing equations are modified, so that the modified equations have a desired analytical solution. The difference to previous studies is that here the technique has been applied for the verification of an unsteady free surface solution method. The verification of such methods has typically been based on i.a. the use of approximate, high order solutions. MMS has the advantage that the numerical solution can be compared with an exact, analytical solution. It is demonstrated in the work that the governing equations were implemented correctly into the developed method and that the method is of second order of accuracy. In addition to the code verification, MMS is used to study the influence of different discretisations and grid refinement strategies on the local error and its convergence. In case of the verification of the free surface solution method the investigation based on a global error norm is extended with an analysis of the Fourier components of the error. A two parameter, approximate model is presented for the temporal variation of the primary component of the solution, with which it is possible to deepen the verification. The model is also used for an uncertainty estimation.

Keywords	Free surface, surface tracking, coupling, code verification, MMS, discretisation error
ISBN (printed) 978-952-248-136-8	ISSN (printed) 1795-2239
ISBN (pdf) 978-952-248-137-5	ISSN (pdf) 1795-4584
Language English	Number of pages 259
Publisher	Teknillinen korkeakoulu, Department of Applied Mechanics
Print distribution	Teknillinen korkeakoulu, Department of Applied Mechanics
☒ The dissertation can be read at http://lib.tkk.fi/Diss/2009/isbn9789522481375/	

VÄITÖSKIRJAN TIIVISTELMÄ	TEKNILLINEN KORKEAKOULU PL 1000, 02015 TKK http://www.tkk.fi

Tekijä Tommi Mikkola

Väitöskirjan nimi

Epästationaaristen vapaan pinnan virtausten simulointi – koodin verifiointi ja diskretointivirhe

Käsikirjoituksen päivämäärä 22.06.2009	Korjatun käsikirjoituksen päivämäärä 30.09.2009
Väitöstilaisuuden ajankohta 06.11.2009	
[X] Monografia	[ ] Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Tiedekunta	Insinööritieteiden ja arkkitehtuurin tiedekunta
Laitos	Sovelletun mekaniikan laitos
Tutkimusala	Laiva- ja meritekniikka
Vastaväittäjä(t)	Professori Luís Eça ja Professori George Tzabiras
Työn valvoja	Professori Jerzy Matusiak
Työn ohjaaja	Professori Jerzy Matusiak

Tiivistelmä

Tässä työssä kehitetään laskennallinen menetelmä ajastariippuvien, kitkattomien vapaan pinnan virtausongelmien ratkaisemiseen, verifioidaan kehitetty menetelmä sekä tarkastellaan yksityiskohtaisesti numeeriseen ratkaisuun liittyvän virheen käyttäytymistä diskretointia tarkennettaessa. Työ jakautuu kahteen erilliseen kokonaisuuteen. Näistä ensimmäinen keskittyy laskentamenetelmän kehittämiseen. Menetelmä perustuu rakenteettomaan, kaksiulotteiseen kontrollitilavuusmenetelmään. Vapaan pinnan reunaehdot toteutetaan hetkellisellä vapaalla pinnalla ja laskentaverkko seuraa pinnan muutoksia. Tyypillisesti vastaavissa menetelmissä virtaus ja vapaan pinnan muutos ratkaistaan aikaintegroimalla vallitsevia yhtälöitä kahdessa toisistaan erotetussa vaiheessa, joita toistetaan vuorotellen. Ratkaisujen erottaminen rajoittaa kuitenkin integroinnissa sallittua aika-askelta, joten lähestymistapa on laskennallisesti varsin raskas. Työssä esitetään kaksi eri tapaa kytkeä virtausratkaisu ja vapaan pinnan ratkaisu, joilla aika-askelrajoitusta saadaan lievennettyä. Esitettyjen tapojen merkittävä ero aiemmin kirjallisuudessa esitettyihin kytkentätapoihin on niiden yksinkertainen implementointi olemassaoleviin painekorjaustyyppisiin ratkaisijoihin. Toisessa osassa keskitytään menetelmän numeerisen toteutuksen oikeellisuuden varmistamiseen eli koodin verifiointiin sekä jatkuvan ongelman diskretointiin liittyvän virheen tarkasteluun. Molemmissa tapauksissa analyysi perustuu ns. MMS-tekniikkaan (method of manufactured solutions), jossa vallitsevia yhtälöitä muokataan siten, että niille saadaan haluttu analyyttinen ratkaisu. Aiemmista, kirjallisuudessa esitetyistä tapauksista poiketen tässä työssä tekniikkaa käytetään epästationaarisen vapaan pinnan ratkaisumenetelmän verifiontiin. Tyypillisesti tällaisten menetelmien verifionti on perustunut mm. likimääräisiin, korkean kertaluvun ratkaisuihin. MMS-tekniikan etu on se, että laskentatulosta voidaan verrata tarkkaan, analyyttiseen ratkaisuun. Työssä osoitetaan, että vallitsevat yhtälöt on implementoitu kehitettyyn menetelmään oikein ja että menetelmä on tarkkuudeltaan toista kertalukua. Verifioinnin lisäksi MMS-tekniikan avulla tarkastellaan eri diskretointien ja verkon tihennysstrategioiden vaikutusta paikalliseen virheeseen ja sen konvergenssiin. Vapaan pinnan ratkaisumenetelmän verifionnin yhteydessä globaaliin virhenormiin perustuvaa tarkastelua laajennetaan virheen Fourier-komponenttien analysoinnilla. Työssä esitetään kaksiparametrinen, likimääräinen malli ratkaisun pääkomponentin aikariippuvuudelle, jonka avulla verifiointia syvennetään. Mallia hyödynnetään myös ratkaisun epävarmuuden arvioinnissa.

Asiasanat	Vapaa pinta, pintaaseuraava, kytkentä, koodin verifiointi, MMS, diskretointivirhe
ISBN (painettu) 978-952-248-136-8	ISSN (painettu) 1795-2239
ISBN (pdf) 978-952-248-137-5	ISSN (pdf) 1795-4584
Kieli Englanti	Sivumäärä 259
Julkaisija	Teknillinen korkeakoulu, Sovelletun mekaniikan laitos
Painetun väitöskirjan jakelu	Teknillinen korkeakoulu, Sovelletun mekaniikan laitos
[X] Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2009/isbn9789522481375/	

Preface

The research presented here has been carried out at the former Ship Laboratory and the current unit of Marine Technology under the Department of Applied mechanics of the Helsinki University of Technology (TKK). The first stage of the research was funded by the Academy of Finland under the "Ship Flows" project. The financial support from the CFD graduate school and my home unit at TKK made it possible to continue this research. These contributions are gratefully acknowledged.

Professor Jerzy (Jurek) Matusiak has been the supervisor and the instructor of this work, but also a mentor and a great support during the odd ten years which I have been working with him. I wish to thank Jurek for the support, guidance and constant encouragement during this period.

D.Sc. Patrik Rautaheimo and Professor Luís Eça have acted as the pre-examiners of this thesis. The valuable comments and critique provided by them is greatly appreciated.

I must also express my gratitude to Mr. Jussi Martio, who first suggested the idea of doing research on unstructured grid based methods. He has also been a great colleague, and the enjoyable on- and off-topic discussions with him have contributed a lot to the presented research.

I also wish to thank Ms. Satu Hänninen and Mr. Pekka Ruponen. It has been a great pleasure to work with them. Satu's work on the further development of the presented method and the related comments and discussions have advanced my research significantly. The discussions with Pekka at the office or in more unofficial circumstances have had the same effect.

The list of people I am grateful to would not be complete without mentioning all the other graduate students of Marine Technology, particularly D.Sc. Jani Romanoff and D.Sc. Heikki Remes. A problem shared is a problem halved. I also wish to thank the full staff of Marine Technology for providing a unique working atmosphere. I am especially grateful to Mrs. Leila Silonsaari for an amazing ability to manage the practical matters at the unit.

The list would not be complete either without my parents, Markku and Helena, my broth-

ers, Markus and Henrik and their families. I wish to thank them for the enormous amount of support and encouragement I have received during all of these years as well as for the "constant interest" they have showed for the work I do. I must also express gratitude to my friends, who have reminded that there is life also outside work.

I have left the most important person last, to make sure that her contribution to this work will not be forgotten. She shared the bad times and the good times, amused me, when I apparently needed it, inspired me with the funny, additional illustrations and comments in the draft versions of the manuscript and will hopefully be a doctor's wife. Jenni, this work is dedicated to you.

Contents

List of abbreviations and symbols

Abbreviations

AIAA	American Institute of Aeronautics and Astronautics
ALE	Arbitrary Lagrangian-Eulerian
CG	Conjugate Gradient
CGSTAB	Conjugate Gradient Stabilized
D-ILU	Diagonal Incomplete Lower Upper
GCI	Grid convergence index
GCL	Geometric conservation law
ITTC	International Towing Tank Conference
MES	Method of exact solutions
MMS	Method of manufactured solutions
MUSCL	Monotone upstream-centred schemes for conservation laws
SAA	Solution accuracy assessment
SCL	Space conservation law
SIMPLE	Semi-implicit method for pressure linked equations
SIMPLEC	SIMPLE, consistent
SQA	Software quality assurance

Greek symbols

α	Matrix coefficient in the pressure correction equation; amplitude factor
α_p	Under-relaxation factor for pressure corrections
α_v	Under-relaxation factor for velocity corrections
β	Phase factor

$\Delta\cdot$	Change of a variable between successive nodes in MUSCL; change of a variable between time steps: superscript specifies, whether the change is related to the pseudo or physical time step; time step
$\delta\cdot$	Difference between quantities at neighbouring cell centres; displacement
$\Delta\tilde{\cdot}$	General measure of the discretisation; approximated variable change
δ_{RE}	The difference between a numerical solution and the Richardson extrapolated value
ε	Specified constant
ε_{ijk}	Permutation symbol
ϕ	General variable
γ	Parameter controlling the blending between first and second order upwind discretisations
η	Wave height
Ω	Domain
$\partial\Omega$	Boundary of Ω
ω	Relaxation factor for grid smoothing; angular frequency
ρ	Density
τ	Pseudo time
θ	Angle between two edges; phase angle

Roman symbols

A	Diagonal term of a matrix; amplitude
a	Matrix coefficient in the implicit free surface equation or momentum equations; arbitrary coefficient; Fourier cosine coefficient
b	Arbitrary coefficient; Fourier sine coefficient
B	Set of points defining boundary geometry

C	Courant number; user defined damping parameter; stiffness of a torsional spring
C	Set of cell centres
d	Damping term
err	Difference between the numerical and manufactured solutions
err_2	Scaled L_2-norm of the discretisation error (see Eq. 5.10)
F	Momentum flux
f	General function
G	Ratio of change of a variable at successive pseudo iterations
g	Acceleration of gravity; coefficient functions for discretisation error (see Eq. 4.1)
h	Free surface coordinates (see Fig. 2.1); discretisation parameter; water depth
I	Interpolation operator: subscript denotes the order
i	Grid level
j	Order of the Fourier component
k	MUSCL switching parameter; stiffness of a linear spring; Jacobi iteration step; wave number
L	Domain length
l	Control volume; cell centre
l'	Auxiliary point in control volume l; arbitrary point in control volume l
$\overrightarrow{ll'}$	Vector from cell centre l to auxiliary point l'
$\overrightarrow{lm}$	Vector from cell centre l to cell centre m
lm	Control volume face; face centre between volumes l and m
$\overrightarrow{lm'}$	Vector from cell centre l to auxiliary point m'
m	Control volume; cell centre; integer constant

m'	Auxiliary point in control volume m
$\dot{m}$	Mass flux
n	Normal vector
$\tilde{N}^{1/10}$	Approximate number of iterations required to drop the L_2-norm of a variable change by one order of magnitude
$n()$	Number of elements in a set
$\tilde{N}^{\infty}$	Approximate number of iterations required to reach converged solution
N_e	Number of elements in a grid
N_{fs}	Number of faces on the free surface
N	Set of vertices
N_T	Number of time steps per one period
N_x	Number of faces on a boundary running along the x direction in the xy coordinate system
N_y	Number of faces on a boundary running along the y direction in the xy coordinate system
p	Pressure; order of accuracy
δp	Skewness correction for pressure
p_t	Order of accuracy of the temporal discretisation
p_x	Order of accuracy of the spatial discretisation
Q	General source term; volume average of a source term
q	Cell vertex; manufactured source term function; cell centre value of a manufactured source term
R	Residual
r	Cross products used for the search of boundary intersection; refinement factor
S	Face of a control volume; size of a face

s	Distance along the free surface (see Fig. 2.1); tangent vector; tangential derivative $\partial/\partial s$
S	Set of control volume faces
T	Residual related to the temporal derivative
t	Physical time
U	Uncertainty
u	The first cartesian velocity component in the xy coordinate system
U_S	Standard error of fit
V	Control volume; size of a control volume
v	Velocity: the components are specified with a subscript; the second cartesian velocity component in the xy coordinate system
$\bar{v}$	Convection velocity
$\delta\bar{v}$	Skewness correction for convection velocity
ΔV	Size of the region swept by a face between physical time steps
v^f	Velocity of the free surface
v_g	Grid velocity
V	Set of control volumes
w	Weight factor
X	Force exerted on a node in spring analogy
x	Position vector: the components are specified with a subscript; the first cartesian coordinate in the xy coordinate system
δx	Distance related to skewness correction (see Fig. 2.6); node displacement in spring analogy
y	The second cartesian coordinate in the xy coordinate system

Subscripts and superscripts

$'$	Auxiliary point; unknown correction

$''$	Unknown skewness corrected correction
$*$	Provisional value after solution of the momentum equations
0	Initial value; specified value; value corresponding to infinite discretisation resolution
1	Quantity related to the start node of a face
2	Quantity related to the end node of a face; order of accuracy based on two discretisation levels
3	Order of accuracy based on three discretisation levels
∞	Quantity related to a fully converged solution
$+$	Value on free surface upwinded on the side of larger s
$-$	Value on free surface upwinded on the side of smaller s
ϕ	Specifies, which variable the quantity refers to
atm	Atmospheric
B_j	Quantity on boundary section j
dyn	Manufactured source term related to the dynamic boundary condition
e	Exact
f	Quantity related to a generic variable f (subscript); quantity related to the free surface (superscript)
high	High order approximation in the deferred correction approach
i	Cartesian vector component; grid level
j	Cartesian vector component; boundary section; free surface node relative to a reference node; order of a harmonic component
k	Point on a boundary geometry definition (subscript); physical time step (superscript)
kin	Manufactured source term related to the kinematic boundary condition

l	Quantity related to the control volume or the cell centre l (subscript); quantity on a face on the side of control volume l (superscript)
l'	Quantity related to auxiliary point for skewness correction in control volume l: the subscript specifies the corresponding face; quantity related to arbitrary point within control volume l
$l \cdot lm$	Quantity at cell centre l projected on face lm
lm	Quantity related to the control volume face or the face centre between volumes l and m; free surface quantity related to face lm; matrix coefficient expressing the influence of quantity at m on quantity at l
$\log(\phi)$	Order of accuracy ov variable ϕ evaluated using the log-log scale
low	Low order approximation in the deferred correction approach
$lp \cdot lm$	Quantity at face centre lp projected on face lm
m	Quantity related to the control volume or the cell centre m; manufactured source term related to the continuity equation (subscript); quantity on a face on the side of control volume m (superscript)
m'	Quantity related to auxiliary point for skewness correction in control volume m: the subscript specifies the corresponding face
ms	Manufactured solution
n	Normal component (subscript); pseudo time step (superscript)
nb	Neighbours of a specified object: the object is specified with a subscript
o	Quantity based on orthogonal approximation
p	Quantity related to vertex or neighbouring cell centre p (subscript); quantity related to the pressure or the pressure correction equation (superscript)
pqr	Triangle formed by nodes p, q and r
pz	Piezometric pressure: the superscript is mostly dropped (see the discussion on page 43)
q	Quantity related to vertex q
qm	Quantity depending on the relation of vertex q and cell centre m

Slm	Quantity averaged over face lm (see Eq. 2.27)
t	Tangential component of a vector
u	Quantity related to the y momentum equation in xy coordinates
v	Quantity related to a vertex: additional subscripts and superscripts specify the vertex; quantity related to the x momentum equation in xy coordinates
Vl	Quantity averaged over control volume l (see Eq. 2.29)

List of Figures

List of Tables

1 Introduction

The work presented here deals with the development of a numerical method for the solution of unsteady free surface flows and with the process of systematically checking the implementation and studying the behaviour of the numerical error of such a method.

1.1 Background

In the case of a fluid flow governed by either the Euler or the Navier-Stokes equations, analytical solutions exist only for a limited set of problems, and generally a numerical solution approach has to be used. Usually, a solution is searched for only for a finite number of locations and instants in time, and a particular variation of the flow variables is assumed between the specified discrete points. The approximation of the continuous problem with a discrete counterpart induces an error, referred to as the discretisation error, and, thus, the computational solution is generally not the same as the exact solution of the governing continuum equations. This error depends on the resolution of the discrete representation of space and time, in addition to the characteristics of the flow. If the numerical method is consistent and stable, the computational solution approaches the solution of the continuous problem as the resolution is increased sufficiently. The manner in which the computational solution converges towards the continuous solution depends on the specific discretisations adopted in the numerical method. However, regardless of the discretisation, it is common that a very large number of discrete points is required to guarantee a sufficiently small discretisation error. In the case of flows around ships, the number of spatial points with a volume discretisation usually ranges between a million and ten million points. Thus it is obvious that for practical purposes the approach has to be implemented as a computer code. The computer code is then used to perform computational simulations. These simulations produce computational solutions to the underlying conceptual model (Oberkampf et al., 2004), i.e. to the governing equations with selected initial and boundary conditions.

For flows involving a free surface the conceptual model describes the flow of a fluid restricted by a deforming material interface. The fundamental problem with free surface flows is that the geometry of the free surface depends on the motion of the fluid, which, on the other hand, depends on the geometry of the free surface. A range of mathematical

models have been devised to tackle this non-linear problem. The earliest of these were based on simplifying assumptions and the linearisation of the problem, so that analytical treatment was possible. A comprehensive review of the subject is presented by e.g. Wehausen and Laitone (1960).

However, the mathematical models that have been developed more recently, at least in the field of ship hydrodynamics, generally take the non-linearity into account exactly. This necessitates the use of a numerical solution approach. One class of methods based on such a model consists of surface tracking methods. The mathematical model for these methods comprises the continuity and Euler or Navier-Stokes equations and boundary conditions, of which those related to the free surface are enforced on the actual free surface. The non-linearity related to the free surface is often treated with an iterative time integration approach, in which the solutions of the flow variables and the free surface deformations are decoupled and the time integration of the corresponding equations is performed alternately (see e.g. Farmer, 1993; Hino et al., 1993; Lungu and Mori, 1993; Tahara and Stern, 1996; Tzabiras, 1997; Löhner et al., 1999; Burg et al., 2002; Apsley and Hu, 2003). Usually, an approximation for the flow field in the next time step is evaluated with the dynamic boundary condition enforcing a specific pressure on the free surface, and then the kinematic boundary condition is used to evaluate a new approximation for the free surface elevation using the updated velocity field. The separate treatment of the flow equations and the kinematic condition yields a restriction on the allowable time step (van Brummelen et al., 2001) and, in this regard, influences the robustness and the convergence characteristics of the method. Because of the limitations of the traditional fully decoupled free surface solution approaches, some authors have proposed alternative coupled approaches.

van Brummelen et al. (2001) coupled the flow and free surface solutions by using a quasi-free surface condition, which takes the influence of the free surface deformation into account in an approximate manner during the flow solution. The condition is based on the expansion of the exact boundary condition relative to a fixed free surface under the assumption that the difference between the actual and the fixed free surfaces is small. The flow solution with the quasi-free surface conditions is iterated until a specified convergence criterion is reached, after which the fixed free surface and the computational grid are updated to conform to the current approximation of the free surface location. This process is repeated until a sufficient convergence for the free surface is reached. The main benefit of the method is that it allows the solution of steady problems without the aforementioned transient approach of the traditional fully coupled methods. This results

in an efficient algorithm. However, as stated by van Brummelen and Segal (2003), the implementation of the quasi-free surface condition can be involved.

Another example of an alternative approach is the fully coupled method presented by (Alessandrini and Delhommeau, 1999). They modified an ill-conditioned and fully decoupled linear system consisting of the momentum and continuity equations and the free surface boundary conditions by expressing the flux through the free surface using the free surface boundary condition. The modification makes the system invertible by an iterative algorithm.

In the field of coating, melting, and solidification technology, fully coupled solution approaches for free and moving boundary problems have been presented by e.g. Sackinger et al. (1996) and Cairncross et al. (2000). These are based on a full Newton strategy and a description of the deformation of the domain with a pseudo-solid approach. This results in a single linear system, which describes the motion of the fluid and the deformation of the free surface and the computational grid at the same time. However, as stated by Sackinger et al. (1996), although robust, "the fully coupled full Newton implementation has the disadvantage of being expensive, compared to other methods, due to the larger system of equations needed to represent mesh displacement unknowns and due to the ill-conditioned matrices that defy most iterative techniques and generally submit only to direct elimination."

In this work, a different viewpoint on the coupling of the solutions has been taken. One of the research questions is whether it would be possible to introduce simple coupling of the bulk flow and free surface solutions into the traditional fully decoupled surface tracking approach without significant changes to the solution method, while still increasing the robustness of the method.

In addition to the robustness of the solution method, other criteria are also used to assess the usefulness of a code for an intended application. One should have confidence that the computational solutions produced by the code adequately represent the physical reality. This means that one should have confidence that the conceptual model is a sufficiently accurate representation of the physical reality, that the conceptual model has been implemented correctly into a computer code, and that the computational solution produced by the computer code is sufficiently close to the solution of the continuous equations of the conceptual model. Oberkampf et al. (2004) state that: "Verification and validation (V&V) of computational simulations are the primary methods for building and quantify-

ing this confidence." The concepts of verification and validation are illustrated in Fig. 1.1,

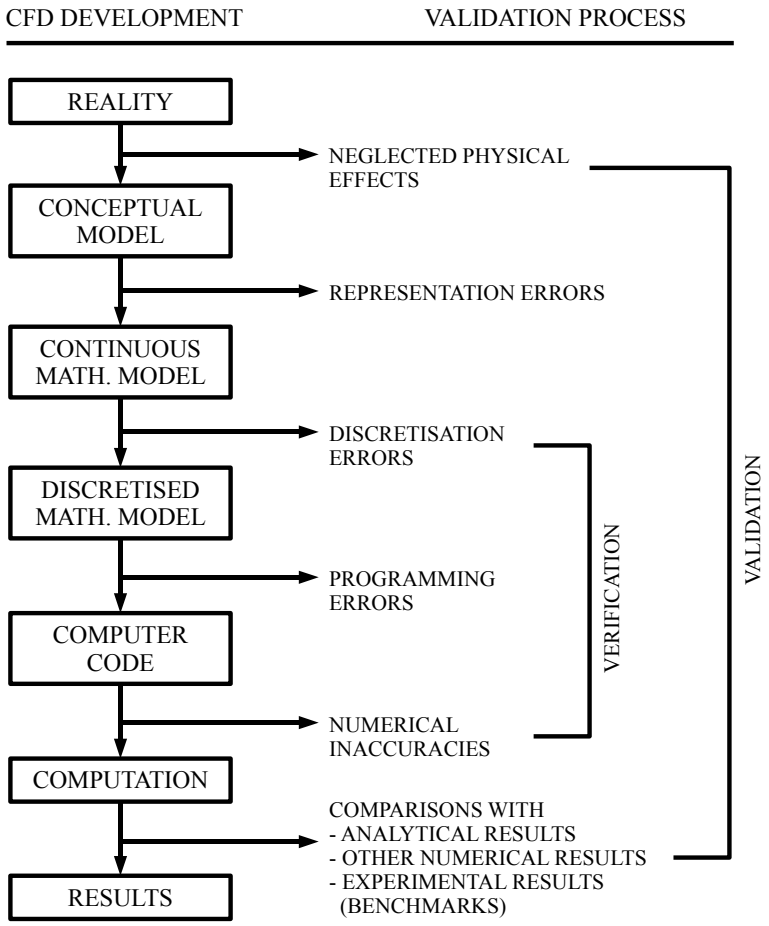


Figure 1.1: Flow diagram for CFD development process. Reproduced from (ITTC, 1990).

which lists the different steps of CFD development and the associated sources of errors as presented by the International Towing Tank Conference (ITTC, 1990). The distinction between verification and validation in computational fluid dynamics can be summarised as (Roache, 1997):

> Verification is concerned with solving the equations right
>
> Validation is concerned with solving the right equations

A somewhat more verbose definition has been adopted by the American Institute of Aeronautics and Astronautics (AIAA) (Oberkampf and Trucano, 2002):

> *Verification*: The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model.
>
> *Validation*: The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

With both definitions, verification is purely a mathematical exercise and deals specifically with the solution of a given set of governing equations representing the conceptual model. It does not deal with the correctness of these equations and, thus, that of the conceptual model in terms of physical laws. The latter is dealt with by validation.

Verification is further divided into two parts: verification of codes and verification of calculations (Roache, 1998). The former deals with the *evaluation* of the error of the computational solution using a known solution to the continuum equations for purposes of comparison. The latter, on the other hand, deals with the *estimation* of the error of a computational solution without the knowledge of the exact solution. To avoid confusion Salari and Knupp (2000) recommend that the term Solution Accuracy Assessment (SAA) is used for the latter.

The exact definitions of the terminology of code verification differ among authors. Therefore it seems necessary to define exactly what is meant by code verification in the context of this work. The definition of code verification by Roache (1998) adopted here is the following:

> "The [code] author defines precisely what continuum partial differential equations and continuum boundary conditions are being solved, and convincingly demonstrates that they are solved correctly, i.e., usually with some order of accuracy, and always consistently, so that as some measure of discretisation (e.g. the mesh increments) $\Delta \rightarrow 0$, the code produces a solution to the continuum equations; this is Verification."

Thus, the purpose of code verification is to demonstrate that the implementation of the conceptual model is correct in the sense that the code is solving the governing equations correctly. Code verification, as defined here, has some relation to Software Quality Assurance (SQA), but does not replace it. The latter is a formal procedure developed to

ensure the reliability and security of software systems (Salari and Knupp, 2000), and focuses on the code as a software product from the point of view of computer science and software engineering (Oberkampf et al., 2004). Furthermore, according to the definition above, code verification does not deal with issues such as the efficiency or the robustness of the numerical method. Roache's definition of code verification is narrower than that presented by e.g. Oberkampf et al. (2004). They argue that code verification should be segregated into two parts, namely, numerical algorithm verification and SQA. However, the numerical algorithm verification they describe has obvious similarities to Roache's definition of code verification. As SQA is outside the scope of this work and the focus is on the numerical performance of the implementation, i.e. on numerical algorithm verification, Roache's definition has been adopted.

Code verification should always precede the verification of calculations (or SAA), which should precede validation, for both logical and practical reasons (Roache, 2002). For a particular code it is sufficient to perform code verification just once, but after modifications the verification has to be repeated (Salari and Knupp, 2000). Additionally, mutually exclusive options of the code have to be verified separately. On the other hand, as the name suggests, the verification of calculations is concerned with a particular solution and should, in general, be performed for each individual calculation.

The obvious requirement of code verification, resulting from the definition presented above, is the knowledge of the exact solution of the continuum equations. However, as already mentioned, because of the non-linear nature of the equations, exact and analytical solutions for the Euler or Navier-Stokes equations exist only in simplified cases. With the free surface included additional complexity is introduced by the non-linear boundary condition on the free surface. Generally, the simplifications required in order to obtain analytical solutions make it impossible to test the implementation of every term of the governing equations and every option of the numerical method. However, the problem related to the lack of analytical solutions can be circumvented by using the Method of Manufactured Solutions (MMS) (for the first uses of the term manufactured solution see Oberkampf et al., 1995; Oberkampf and Blottner, 1998; Reed et al., 1998). Here the governing equations are modified with source terms in such a way that a known, exact, analytical solution exists for the modified equations. MMS combined with discretisation refinement, first presented by Steinberg and Roache (1985), is a systematic procedure for code verification.

The applicability of MMS is not restricted to a specific field of computational science

and engineering, and the method has been used extensively in different fields, including computational fluid dynamics. The historical references for the application of MMS to code verification have been cited and reviewed in several recent papers and books (e.g. Roache, 1997, 1998, 2002; Salari and Knupp, 2003; Roy, 2005). More recent examples of the use of MMS include the comprehensive studies performed within the realm of the 2^{nd} and the 3^{rd} Workshop on CFD Uncertainty Analysis (Eça and Hoekstra, 2006a, 2008). The organisers of the workshops devised manufactured solutions mimicking a turbulent flow over a flat plate and derived the corresponding source terms for different turbulence models. The manufactured solution was used by the participants to verify their codes, but also to assess the reliability of different uncertainty estimators. A related paper focusing on the same test cases as the workshops has been published by Eça et al. (2007). Other examples of MMS-based code verification studies include groundwater-related studies for methods solving porous media equations. An unsteady manufactured solution for this purpose has been presented by Salari and Knupp (2003), and Burg and Murali (2004) presented a code verification study using a radially symmetric flow. Murali and Burg (2002) also applied MMS for the code verification of a two-dimensional, unstructured, Euler equation solver on grids composed of triangles. A similar study with the verification of both an unstructured and a structured solution method using the Euler and Navier-Stokes equations has been presented by Roy et al. (2004). Furthermore, Deng et al. (2006) present a code verification of an unstructured solver with three different grid types using MMS.

Despite the large number of examples of code verification studies in computational fluid dynamics, systematic code verification studies with unsteady free surface flows are particularly lacking. However, some examples of verification studies related to the simulation of surface waves can be found in the open literature, although the method of manufactured solutions was not used in them. Chen et al. (1999) studied the grid convergence of a volume of fluid-type free surface solver by simulating capillary and gravity waves. The code verification was based on a comparison of the frequency of the simulated wave and the theoretical predictions as the grid is refined. Hu and Kashiwagi (2004) used the propagation of a solitary wave for the verification of their constrained interpolation profile-based free surface solver. The consistency of the discretisation was verified qualitatively by studying the attenuation of the maximum wave height of the propagating wave with grid refinement. Hur et al. (2008) used a third-order Stokes wave to verify the direct numerical simulation-based volume of fluid solver for the fully non-linear simulation of free surface flows. The verification study is restricted to a qualitative comparison of the simulated and theoretical wave elevations with grid convergence. What is common to all of these

examples is the fact that the studies are mainly qualitative. Qualitative here refers to a graphical comparison of values, with consistency demonstrated by a visual convergence of the simulation result towards the assumed exact solution of the governing equations.

1.2 The scope of the work

In this work, the scope is restricted to Finite Volume (FV) methods, in which the domain of interest is divided into small sub-domains, referred to as control volumes. This restriction is based on the fact that the majority of the volume methods currently in use in the field of ship hydrodynamics are based on FV discretisation (see e.g. Hino, 2005). A further restriction is made by only considering surface tracking methods in two dimensions with the assumption of inviscid flows. However, it is believed that despite these simplifications some of the fundamental properties and issues of the time-accurate solution of unsteady free surface flows – particularly related to the solution and the error behaviour of propagating waves – are revealed. Moreover, from the point of view of verification and validation the focus in this work is on code verification and the verification of calculations, whereas the validation of the method is intentionally left outside the scope of this work and has been addressed elsewhere (see e.g. Mikkola, 2006). In the context of unsteady free surface flows verification – and particularly code verification – is a largely unexplored area. However, the importance of the topic is underlined by the increasing number of unsteady simulations (e.g. ships in waves) performed in the field of ship hydrodynamics (see e.g. Hino, 2005; Carrica et al., 2006). This was the fundamental motivation for this work.

The main contributions of the work presented are listed below.

1. A second-order accurate unstructured surface tracking method for unsteady free surface flows has been developed (see Chapter 2). The method has been developed independently by the author and the implementation was originally discussed in (Mikkola, 2002, 2003, 2004, 2005, 2006). The method is intended to be a research platform and was later extended with a VOF-type surface capturing capability (Hänninen and Mikkola, 2007).

2. In an attempt to improve the robustness of the traditional fully decoupled surface tracking solution approach, simple coupling of the bulk flow and free surface solutions is proposed (see Sec. 2.6). This exploits the segregated nature of the solu-

tion algorithm resulting from the pressure-velocity decoupling adopted in pressure correction-type methods. The coupling has been developed independently by the author and was first presented in (Mikkola, 2003). The approach can be implemented in a straightforward fashion in existing pressure correction-type solvers.

3. A code verification study using the method of manufactured solutions has been performed to analyse the grid convergence of different unstructured spatial discretisations and two different unstructured grid refinement strategies (see Chapter 5). The global error norm-based studies were supplemented by a detailed analysis of the distributions of local error. A subset of this study was first presented in (Mikkola, 2007).

4. The application of the method of manufactured solutions for the code verification of an unstructured and time-accurate surface tracking free surface flow solver has been demonstrated (see Chapter 6). In addition to the global error norm-based study, an approximation of the primary component of the discretisation error was constructed in order to study the temporal behaviour of the error and to allow a more detailed quantitative verification of the free surface solver. This quantitative study, first presented in (Mikkola, 2007), was supplemented by a qualitative analysis of the spatial and temporal variation of the local error of both the primary and the secondary error components.

The full research and the reporting of this work was performed independently by the author. The supervisor and the instructor of the thesis or any other individual contributed to the work only through discussions and by providing comments on the manuscript of the thesis.

1.3 The structure of the theses

In Chapter 2 a detailed description of the solution method is given. First, the governing equations are presented, and this is followed by a brief overview of the unstructured finite volume method. The bulk of the chapter consists of a description of the implementation of the method. The discussion covers the spatial and temporal discretisations, the construction of the system of algebraic equations and the corresponding solution algorithm, the discretisation of the boundary conditions, and the grid-updating algorithm.

The performance of the coupling of the flow and free surface solutions is studied and discussed in Chapter 3. Three free surface test cases, the steady flows over a ground elevation and around a submerged hydrofoil and the unsteady flow of a standing wave, were used. The chapter is partly based on a paper presented by the author (Mikkola, 2003).

Chapter 4 gives an overview of code verification with the method of manufactured solutions. Two aspects of the topic are covered. First, the different sources of numerical error are discussed. After this, the motivation and the basic idea of the method of manufactured solutions are presented.

The code verification studies are performed in two parts. First the implementation of the bulk flow solution in the method that was developed is verified in Chapter 5 using a steady flow without a free surface by Salari and Knupp (2000) as the manufactured solution. Different discretisation options and two different grid refinement strategies are studied. Both quantitative and qualitative verification studies were performed. The behaviour of the numerical error with grid refinement was analysed, not only in terms of global error norms, but also in terms of the detailed distributions of local error. In Chapter 6 the time-accurate solution method for free surface flows is verified using the linearised potential flow solution of a standing wave in a rectangular container as the manufactured solution. The verification is based on the analysis of the behaviour of a global error norm and the Fourier components of the simulated wave with the simultaneous refinement of the spatial and the temporal discretisations. The construction of an approximation for the temporal variation of the primary component of the simulated wave is discussed. The approximation is applied for a more detailed quantitative verification of the method that was developed. The validity of the approximation is also studied. The behaviour of the local and the secondary components of error with discretisation refinement is studied qualitatively.

The main conclusions of the code verification studies are discussed in Chapter 7. The results are discussed from two points of view. First, the verification of the consistency and order of accuracy of the code implementation is discussed. Second, the influence which the discretisation error has on the behaviour of the local error is covered.

Chapter 8 summarises the thesis. It draws conclusions from the whole work and gives recommendations for future work.

Part I

Method development

2 The numerical method

The discussion of the solution method starts with a definition of the governing equations (Sec. 2.1), followed by a brief overview of the unstructured finite volume method (Sec. 2.2). After these, a detailed description of the solution method – divided into seven sections – is given. The first two sections (Secs. 2.3 and 2.4) deal with the spatial and temporal discretisations, respectively, i.e. with the discrete approximation of the governing equations. These are followed by sections on the construction of the algebraic equations and on the approach to the solution of these equations for the bulk flow and the free surface deformation, respectively (Secs. 2.5 and 2.6). After these, the discrete formulation of the boundary conditions is discussed (Sec. 2.7). The description of the solution method is concluded with a section on the grid-updating algorithm (Sec. 2.8).

2.1 The governing equations

The flow is assumed to be incompressible, inviscid, and isothermal in 2D. A flow with density ρ through a region Ω is considered. The region is defined by an arbitrary bounding surface $\partial\Omega$, which has an outer normal vector n_i. Here, i refers to the ith component of the vector. In this section, the discussion of the governing equations has been divided into two subsections. The aim is to highlight the influence which the presence of the free surface has on the governing equations. In the first section the arbitrary domain Ω is assumed to be fixed, i.e. neither the shape nor the location of the domain changes with time. The governing equations and the associated differences for free surface bounded flows with a deforming domain Ω are then discussed in the second section. These sections are followed by a discussion of the most fundamental aspect of free surface flow equations, namely the free surface boundary conditions.

2.1.1 The equations for fixed domains

The unknowns to be solved are the Cartesian velocity components v_i and the pressure p. In this thesis the implied summation over repeated indices in a term is used and, thus,

$$v_i n_i = \sum_i v_i n_i \ . \tag{2.1}$$

Later, additional indices referring to the elements in the discretised form of the governing equations will be introduced. The implied summation applies only to the indices which refer to vector components and therefore does not apply to the discrete elements.

The governing equations for the flow are the incompressible continuity and Euler equations in the conservation form. In problems of fluid mechanics the Eulerian description is generally preferred over the Lagrangian one. This has also been done in the method that was developed. Thus, the flow variables are defined as properties of the domain rather than as properties of particles. However, the governing equations are based on the conservation of the properties of a particle system. The transformation between these two systems is based on the Reynolds transport theorem (Donea et al., 2004)

$$\frac{d}{dt}\int_{\Omega}\rho\phi dV = \frac{\partial}{\partial t}\int_{\Omega}\rho\phi dV + \int_{\partial\Omega}\rho v_i v_j n_j dS\ . \tag{2.2}$$

The temporal derivative d/dt is the material derivative. This is defined as the time rate of change of a general property ϕ associated with the system of particles. This general property has a density $\rho\phi$.

The continuity equation

$$\frac{d}{dt}\int_{\Omega}\rho dV = 0 \tag{2.3}$$

is based on the conservation of mass. Here, the integral is equal to the mass of the system of particles occupying the domain Ω at a given instant. By substituting Eq. (2.2) into Eq. (2.3) with $\phi = 1$ and by taking into account the fact that Ω does not change and ρ is constant, the incompressible continuity equation can be written as

$$\int_{\partial\Omega}\rho v_i n_i dS = 0\ . \tag{2.4}$$

This states that the inflow of mass into the domain has to equal the outflow of mass from the domain. If the convection velocity is defined as $\bar{v} = v_i n_i$, the continuity condition is given as

$$\int_{\partial\Omega}\rho\bar{v}dS = 0\ . \tag{2.5}$$

In addition to the conservation of mass, the balance of momentum – expressed by the Euler equations for inviscid fluid – has to be satisfied in the flow. For the incompressible flow free of external forces the balance can be written as

$$\frac{d}{dt}\int_{\Omega}\rho v_i dV = -\int_{\partial\Omega} p n_i dS\ . \tag{2.6}$$

The left-hand side is equal to the time rate of change of the momentum of the mass system occupying the domain Ω at a given instant. The right-hand side represents the force exerted on the domain through its boundary $\partial\Omega$. By using Eq. (2.2) with $\phi = v_i$ the equation is given with Eulerian description as

$$\frac{\partial}{\partial t}\int_\Omega \rho v_i dV + \int_{\partial\Omega} \rho v_i v_j n_j dS = -\int_{\partial\Omega} p n_i dS \,. \tag{2.7}$$

Now the left-hand side consists of the time rate of change of the momentum within the fixed domain Ω and the net outflow of momentum through the boundary of Ω. Eq. (2.7) leads to two equations in 2D, one for each velocity component. Noting the convection velocity in the second integral on the left-hand side, the momentum balance may be expressed in the form

$$\frac{\partial}{\partial t}\int_\Omega \rho v_i dV + \int_{\partial\Omega} \rho v_i \bar{v} dS = -\int_{\partial\Omega} p n_i dS \,. \tag{2.8}$$

The final forms of the governing equations for a fixed domain are, thus, Eqs. (2.5) and (2.8). The governing equations in the case of a changing domain are discussed next.

2.1.2 The equations for deforming domains

If a free surface or some other moving boundary is present in a flow problem, the shape of the computational domain changes with the deformation and translation of such boundaries. Therefore, the assumption of fixed Ω made above is not valid and the governing equations have to be modified. In the method that was developed the arbitrary Lagrangian-Eulerian (ALE) description was used. This is a cross between the Lagrangian and Eulerian descriptions.

In ALE, the grid can be fixed to the space as in Eulerian description; it can move by following the motion of the fluid particles, as in Lagrangian description, or it can move arbitrarily, regardless of the motion of the fluid. In the surface tracking method that was developed the displacements of the nodes of the computational grid are dictated by the deformation and translation of the free surface and other moving boundaries. The node displacements in the rest of the domain vary continuously between the displacements at the moving boundaries and zero displacement for stationary nodes far away from the moving boundaries. The grid deformation algorithm is discussed in more detail in Sec. 2.8. For further discussion on the ALE description in general the interested reader is referred to e.g. Donea et al. (2004).

Instead of the governing equations being written for a fixed Ω, they now have to be written for a deforming and translating domain $\Omega(t)$. For this the general Reynolds transport theorem for arbitrarily moving domains (Ferziger and Perić, 1997; Donea et al., 2004)

$$\frac{d}{dt}\int_{\Omega(t)} \rho\phi dV = \frac{\partial}{\partial t}\int_{\Omega(t)} \rho\phi dV + \int_{\partial\Omega(t)} \rho\phi\bar{v} dS - \int_{\partial\Omega(t)} \rho\phi v_g dS \tag{2.9}$$

has been used. Here, v_g is the normal component of the velocity of the boundary of the domain Ω. Using Eq. (2.9) the continuity equation (2.3) for an arbitrarily moving domain $\Omega(t)$ can be written as

$$\frac{d}{dt}\int_{\Omega(t)} \rho dV = \frac{\partial}{\partial t}\int_{\Omega(t)} \rho dV + \int_{\partial\Omega(t)} \rho\bar{v} dS - \int_{\partial\Omega(t)} \rho v_g dS = 0\ . \tag{2.10}$$

The equation also has to be satisfied in the trivial case that the fluid is stationary, i.e. when the velocity $\bar{v}$ is uniformly zero. This results in the so-called geometric or space conservation law (GCL or SCL, Hoffren, 1993; Ferziger and Perić, 1997)

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \rho dV = \int_{\partial\Omega(t)} \rho v_g dS\ . \tag{2.11}$$

For a constant density this simplifies to

$$\frac{\partial}{\partial t}\int_{\Omega(t)} dV = \int_{\partial\Omega(t)} v_g dS\ . \tag{2.12}$$

With Eq. (2.11) Eq. (2.10) reduces into

$$\int_{\partial\Omega(t)} \rho\bar{v} dS = 0\ , \tag{2.13}$$

which is simply Eq. (2.5) applied to the instantaneous domain $\Omega(t)$. Thus, continuity can be treated in exactly the same way, regardless of the deformation of the computational domain.

The momentum equations are treated in a similar fashion, i.e. the momentum equations (2.6) are written for an arbitrarily moving domain $\Omega(t)$ and the general Reynolds transport theorem (2.9) is used for the material derivative. For free surface flows gravity has a significant role and its influence has to be included in the governing equations. Therefore, an additional body force term appears on the right-hand side of the momentum equations. With these modifications, Eq. (2.8) with a changing domain, $\Omega(t)$ becomes

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \rho v_i dV + \int_{\partial\Omega(t)} \rho v_i \left(\bar{v} - v_g\right) dS = -\int_{\partial\Omega(t)} p n_i dS + \int_{\Omega(t)} \rho g_i dV\ , \tag{2.14}$$

where g_i are the components of the gravity vector. By applying the Gauss theorem for the gravity term and taking into account the fact that gravity and density are constants, Eq. (2.14) can be written as

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \rho v_i dV + \int_{\partial\Omega(t)} \rho v_i \left(\bar{v} - v_g\right) dS = -\int_{\partial\Omega(t)} p n_i dS + \int_{\partial\Omega(t)} \rho g_i x_i n_i dS \; . \quad (2.15)$$

By introducing the piezometric pressure given by

$$p^{pz} = p - \rho g_i x_i \; , \quad (2.16)$$

and by substituting this definition in Eq. (2.15), the momentum equations become

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \rho v_i dV + \int_{\partial\Omega(t)} \rho v_i \left(\bar{v} - v_g\right) dS = -\int_{\partial\Omega(t)} p^{pz} n_i dS \; . \quad (2.17)$$

Now the right-hand sides of the momentum equations have exactly the same form as Eq. (2.8), also in cases involving the influence of gravity. Therefore, p is used in the following sections as a general notation for pressure, and, depending on the case, it may refer to the total pressure or the piezometric pressure. As a general rule, if a case includes gravity, p refers to the piezometric pressure; otherwise it refers to the total pressure.

2.1.3 The free surface boundary conditions

The bulk flow and free surface are connected through boundary conditions, which have to be satisfied on the deforming surface. As a material interface, the free surface introduces two types of conditions on the flow quantities at the interface.

The kinematic boundary condition states that there is no flow through the interface. This requires that

$$\left(v_i - v_i^f\right) n_i = 0 \; , \quad (2.18)$$

where v_i^f are the components of the velocity of the surface. In free surface methods it is quite common that for the sake of simplicity the velocity of the surface is defined as being vertical over the whole free surface. However, in the method that was developed the free surface deforms in the direction of the surface normal. The free surface is defined with a parametric curve $h_i(s)$, where s is a parameter measuring the distance along the curve from the right-hand end of the boundary (see Fig. 2.1). The velocity of the surface is given by

$$v_i^f = \frac{\partial h_i}{\partial t} \; . \quad (2.19)$$

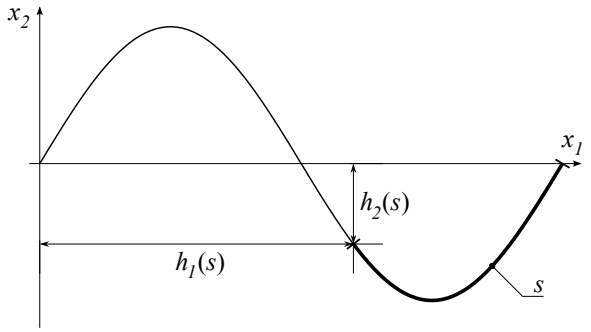


Figure 2.1: The parametric representation of the free surface. The parameter s is the distance along the curve from the right-hand end of the curve.

Eqs. (2.18) and (2.19) are not sufficient to describe the deformation of the free surface, as there are two unknowns (h_1 and h_2) and just one condition. Thus, the direction of the surface deformation can be chosen freely and in this case the kinematic boundary condition (2.18) only restricts the length of the displacement vector. In this work the deformation of the free surface is defined to be in the direction of the free surface normals, giving the required additional relation

$$\frac{\partial h_i}{\partial t} = \frac{\partial h_n}{\partial t} n_i \; . \tag{2.20}$$

By substituting Eqs. (2.19) and (2.20) into Eq. (2.18) the final form for the kinematic boundary condition is reached, i.e.

$$\frac{\partial h_n}{\partial t} = v_i n_i \; . \tag{2.21}$$

For a steady state solution the surface velocity v_i^f vanishes and the kinematic boundary condition reduces to

$$v_i n_i = 0 \; . \tag{2.22}$$

The second condition to be satisfied on the free surface is the dynamic boundary condition. This states that stresses have to be continuous across the free surface. In this work, the inviscid approximation of this without surface tension effects is used. With these approximations the dynamic boundary condition reduces into a condition for the normal stress, giving

$$p = p^{atm} \; . \tag{2.23}$$

Some additional assumptions are also made. First, zero atmospheric pressure p^{atm} is assumed. Second, throughout this work it is assumed that gravity points in the negative x_2 direction, i.e. the gravity vector is $(0, -g)$, where g is the absolute value of gravitational

acceleration. Thus, taking into account Eq. (2.16), the dynamic boundary condition for the piezometric pressure p^{pz} on the free surface can be written as

$$p^{pz} = \rho g h_2 \, . \tag{2.24}$$

2.2 Unstructured finite volume method

It should be noted that the domains Ω and $\Omega(t)$ in Sec. 2.1 are arbitrary and no assumptions have been made regarding either the shape or the location of the domains. Thus, Eqs. (2.5) and (2.8) or (2.13) and (2.17) are fully general and are valid for domains of any shape, size, and location. In the finite volume method the domain of interest is divided into a finite number of non-overlapping sub-domains, i.e. control volumes. The method used for the solution of the bulk flow is based on the discretisation of the domain of interest into such control volumes and on the application of the equations in conservation form for each individual control volume.

The resulting control volume grids can be divided into two major classes: structured and unstructured. Structured grids have a well-defined structure, from which the neighbours of a particular volume can be readily deduced on the basis of the location of the volume in the structure. If unstructured grids are used, the volumes may be ordered arbitrarily. However, the connections between neighbours have to be defined explicitly with dedicated data structures, as it is not possible to deduce the indices of the neighbouring cells from the structure of the grid. The latter approach was adopted in the method that was developed.

Both structured and unstructured grid-based methods have their respective advantages and disadvantages. In the case of structured grids the relations between neighbouring elements are known, which simplifies the construction and implementation of discretisations significantly, particularly for high-order discretisations involving large discretisation stencils. Furthermore, as a result of the well-defined, banded nature of the linear equation systems arising from the use of structured grids, the numerical solution methods for these are very efficient. For unstructured grids the neighbour relations are arbitrary and the construction of high-order discretisations is much more complicated. More general linear system solvers also have to be used, as no assumptions can be made about the exact structure of the linear system. On the other hand, grid generation, especially for complex geometries, has become a bottleneck in the solution process using structured methods.

The advantage of the unstructured approach lies particularly in the flexibility of the grid generation. Further discussion of the characteristics of the different approaches can be found in e.g. (Demirdžić et al., 2000).

The fundamental idea in the finite volume method is that the integral equations (2.5) and (2.8) – representing the conservation of mass and momentum – are applied for each control volume separately. Volume integrals are therefore integrals over each control volume and the boundary integrals become integrals over the faces of the volume. Fig. 2.2 shows a general finite volume discretisation with the definition of the notations for the volumes and faces. The general notation convention is as follows. Quantities related to a particular volume or cell centre are referred to with a single index – usually l or m. Quantities related to a particular face or face centre are referred to with a double index – usually lm – with the two indices designating the cells on either side of the face. Even if in this work only two-dimensional flow problems are considered and the cells of the computational grid are surface elements, the cells are regularly referred to as volumes and denoted by V. The size of a control volume is also denoted by V. Similarly, the sections of the boundary of a control volume are referred to as faces, and the face and the corresponding dimension are denoted by S, even if in this work they are one-dimensional objects.

For an arbitrary polygonal control volume l the conservation of mass (2.5) can be written as a sum over the faces $lm \in \mathrm{S_l}$ of the volume

$$\sum_{lm} \dot{m}_{Slm} = 0, \quad lm \in \mathrm{S_l}\ . \tag{2.25}$$

Here

$$\dot{m}_{Slm} = \rho \bar{v}_{Slm} S_{lm} \tag{2.26}$$

is the mass flux through the face lm connecting the control volumes l and m and

$$\bar{v}_{Slm} = \frac{1}{S_{lm}} \int_{lm} v_i n_i dS \tag{2.27}$$

is the average convection velocity over the face lm.

Similarly, the balance of momentum (2.8) for an arbitrary control volume l with size V_l can be written as

$$\frac{\partial \rho V_l v_{i,Vl}}{\partial t} = -\sum_{lm} F_{i,Slm}, \quad lm \in \mathrm{S_l}\ . \tag{2.28}$$

Here $v_{i,Vl}$ in Eq. (2.28) are the velocity components averaged over the control volume l

$$v_{i,Vl} = \frac{1}{V_l} \int_l v_i dV\ . \tag{2.29}$$

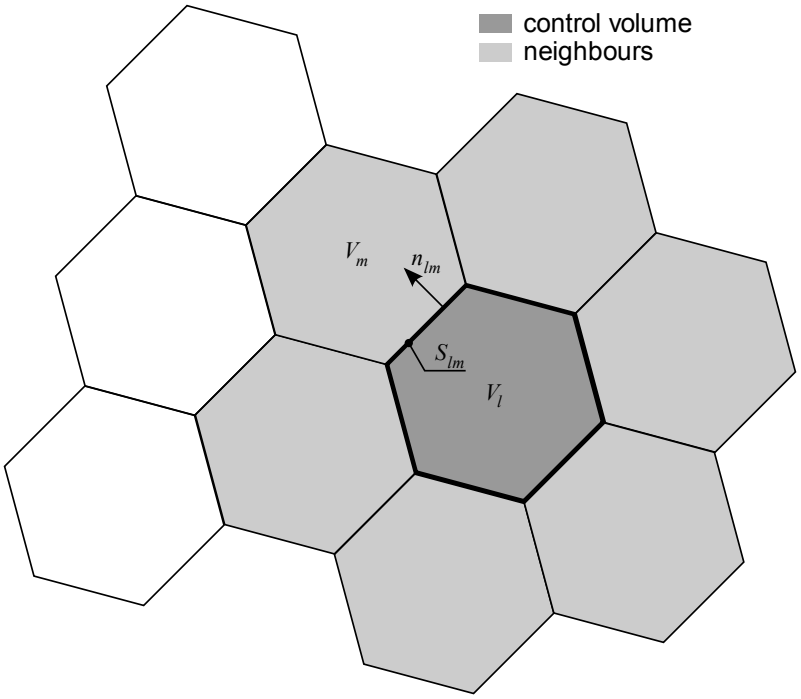


Figure 2.2: A general finite volume discretisation with the definition of control volumes, neighbours, and faces. The neighbours of a control volume $V_l \in \mathrm{V}$ are defined as being the set of volumes $\mathrm{V}_\mathrm{l}^\mathrm{nb} \subset \mathrm{V}$, which share at least one face with the control volume. The set of the common faces $\mathrm{S}_\mathrm{l} \in \mathrm{S}$ corresponds to the boundary of the control volume V_l.

The term $F_{i,Slm}$ on a face lm is given by

$$F_{i,Slm} = [\rho(v_i \bar{v} - v_i v_g)_{Slm} + (p n_i)_{Slm}]\, S_{lm} \; . \tag{2.30}$$

For simplicity of presentation, the term (2.30) as a whole is referred to as the momentum flux in this work, even if the pressure term is not an actual flux, but a force exerted on the face. The velocity and pressure terms $(v_i \bar{v} - v_i v_g)_{Slm}$ and $(p n_i)_{Slm}$ in the flux (2.30) are averaged values over the face lm, similarly to the average in Eq. (2.27).

By looking at Eqs. (2.26) and (2.30) it can be seen that in order to calculate the fluxes, the averages of the velocity components and pressure on the control volume faces are required. Furthermore, the volume average is needed as well. The evaluation of these quantities is an integral part of a finite volume method and depends largely on the discretisation formulation, e.g. whether a cell-centred or vertex-based method is used and whether the discretisation is structured or unstructured. Additionally, a discrete approximation for the time derivative is required (see Eq. 2.28).

The details of these for an unstructured cell-centred scheme will be discussed next. First, the spatial discretisation and the associated construction of the fluxes will be covered. This is followed by discussion of the temporal discretisation, i.e. the discrete approximation of the time derivative.

2.3 Spatial discretisation

The finite volume discretisation of the governing equations results in a system of equations, in which the equations for each control volume have to be satisfied simultaneously. Fluxes on the faces of the volumes are required for the continuity condition (2.25), as well as for the momentum equations (2.28). The solvability of the system depends on the relation between the number of unknowns and the number of independent equations in the system. In order to have a finite set of solutions, the number of unknowns has to be equal to the number of equations in the system. As the number of equations is proportional to the number of control volumes, the choice of the unknowns and the control volumes are linked. These unknowns are then used to approximate the volume and face averages in the equations.

In the following sections a full description of the discretisation of the face values is given. First, the general aspects of the spatial discretisation of the domain and the boundaries are discussed. After this general part, attention is turned to the practical implementation of the unstructured finite volume scheme in the solution method that was developed, and the actual construction of the flux terms on the faces is discussed in more detail.

2.3.1 Collocated cell-centre discretisation

The two most common approaches to spatial discretisation with unstructured grids are cell-centred and cell-vertex discretisations. In the cell-centred approach the cells of the computational grid are taken as the control volumes and the unknowns are the values at the centres of the cells. In the cell-vertex approach the values at the vertices of the computational grid are taken as the unknowns and the control volumes are constructed separately around the vertices.

In the method that was developed a collocated, cell-centred discretisation was used. The

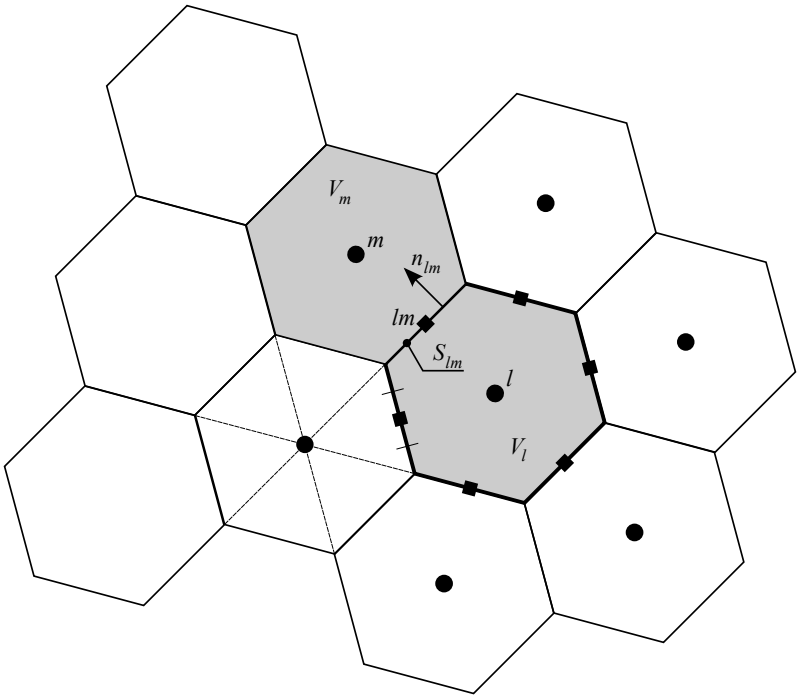


Figure 2.3: A cell-centred finite volume discretisation with the definition of cell centres (circles) and face centres (squares). The control volumes $V_l, V_m \in \mathrm{V}$ (shaded) associated with the construction of the flux on face $S_{lm} \in \mathrm{S}$ are shown as well. The set of cell centres which neighbour the cell centre $l \in \mathrm{C}$ is denoted with $\mathrm{C}_\mathrm{l}^\mathrm{nb} \subset \mathrm{C}$.

term 'collocated' here refers to the fact that the discrete velocities and pressures are defined at the same locations. Fig. 2.3 gives a general overview of cell-centred discretisation. For the sake of brevity, in the following sections the same indices (e.g. l or lm) are often used to refer to both the control volume (V_l) and the corresponding cell centre (l) or to both the control volume face (S_{lm}) and the corresponding face centre lm. To avoid confusion, the exact meaning of the reference is clearly stated, e.g. volume l or cell centre l and face lm or face centre lm.

The volume average of a variable (2.29) is approximated with the value of the variable at the geometric centre l of the control volume, i.e. directly with the unknowns of the scheme. Thus, for velocity components the approximation is

$$v_{i,Vl} = v_{i,l} \ . \tag{2.31}$$

This approximation based on the midpoint rule is second-order accurate (Demirdžić et al., 2000).

In addition to the volume averages, approximations for the face averages over the control volume faces, such as (2.27), are required for the evaluation of the fluxes. Following the approach used for the volume averages, the face averages are also approximated using the midpoint rule, i.e.

$$v_{i,Slm} = v_{i,lm} \ . \tag{2.32}$$

For this, the value at the face centre lm is required. The fundamental difference between the approximations of the volume and the face averages is that for the face average the midpoint value is not directly available. Therefore, it has to be approximated using the unknowns of the scheme. In a cell-centred scheme the face centre values are approximated using the values at the cell centres surrounding a face.

2.3.2 Discretisation of the boundaries

For faces at the boundaries of the domain a special treatment of the face centre values is required, as there are cell centres only on one side of a boundary face. There are two common approaches that are used to deal with boundary values in finite volume formulations. In the first approach, a separate treatment is used for fluxes on faces which lie at the boundary of the computational domain, i.e. the boundary fluxes and the fluxes for the faces inside the domain are treated differently. In the second approach a uniform treatment is used for all the faces. Here the values on the outside of the boundary, required by the uniform flux evaluation scheme, are extrapolated on the basis of the boundary conditions and are stored in so-called ghost cells. With the data structures available in the method that was developed both approaches are easily adopted, but the latter approach was opted for. In this section some general aspects of the ghost cell approach are presented. Additionally, the definition and discretisation of the boundary geometry are described. The exact implementation of the boundary conditions is discussed later in Sec. 2.7.

In the approach used, there is one ghost cell for each face at the boundary. In most cases, the ghost cells are created by mirroring the cell centre of the control volume, which is next to the boundary face, relative to the centre point of the face. However, those sections of the boundary with a symmetry condition are an exception. For these the ghost cell centre is a mirror image of the computational cell centre relative to the boundary. The two alternatives are illustrated in Fig. 2.4.

In the method that was developed, the ghost cells are not valid control volumes. The actual geometry of a ghost cell, in terms of the vertex coordinates of the cell, is neither

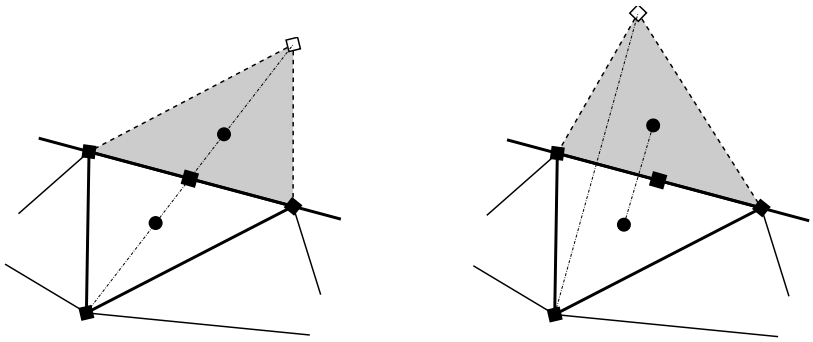


Figure 2.4: Ghost cells (shaded) outside the boundary: normal ghost cell on the left and ghost cell for the symmetry condition on the right.

used nor available. Only the cell centre values, such as flow variables and the location of the cell centre, as well as the size of the ghost cell copied from the computational domain, are stored in memory.

The parametric free surface curve $h_i(s)$ is discretised by defining one wave height node $h_{i,lm}$ for each boundary face. With this definition the application of the free surface boundary conditions (2.18) and (2.24) is a straightforward task, as now the boundary conditions involve face centre velocities and pressures, similarly to the flux terms.

The boundary faces are special not only in terms of flux evaluation, but also because the geometry of the computational domain is specified by the piecewise linear path formed by these outermost faces of the grid. For cases in which the geometry of the domain does not change with the solution, this is a sufficient description of the boundary geometry. However, if this is not the case and the geometry changes, a more complete and accurate description of the boundary is required in order to translate the boundary nodes along the geometry. The geometry of a boundary section j is defined using a separate piecewise linear curve $\mathrm{B_j}$, along which the nodes of the boundary section are displaced. This is illustrated in Fig. 2.5.

In the case of a boundary section with a known geometry, the definition curve $\mathrm{B_j}$ is generated in the pre-processing step using a sufficient number of nodes $x_i^{B_j}$ to give an accurate description of the real geometry. The free surface sections of the boundary are treated differently, as the shape of the surface is not known beforehand and a predefined geometry cannot be used. The geometry of the free surface is defined with the piecewise linear

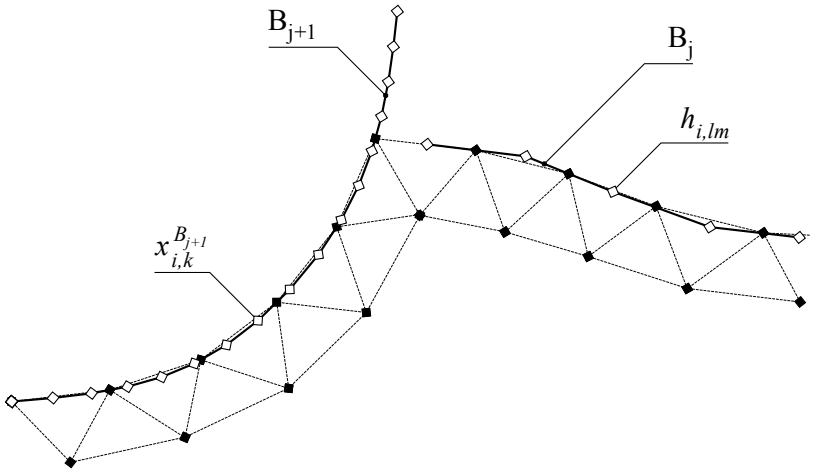


Figure 2.5: The definition of the boundary geometry. The piecewise linear curves $\mathrm{B_j}$ defined with the nodes $x_i^{B_j}$ (open diamonds) represent the geometry of the boundary along which the nodes of the computational grid (solid diamonds) are displaced. In the figure section j is a free surface boundary and section $j+1$ is a section with a predefined geometry.

curve $\mathrm{B_j}$ connecting the wave height nodes h_i (see Fig. 2.5), i.e.

$$x_{i,k}^{B_j} = h_{i,lm} \; . \tag{2.33}$$

During the solution process the wave height nodes are updated according to the deformation of the free surface, and the updated definition of the boundary geometry is used to displace the grid nodes. The details of this are discussed in Sec. 2.8.

2.3.3 Momentum flux

The flux in Eq. (2.30) consists of two parts, the convection part and the pressure part. As discussed before in Sec. 2.3.1, the approximation of the face averages in the fluxes is based on the midpoint rule. Thus, the average of a quantity ϕ_{Slm} is represented by the corresponding value ϕ_{lm} at the centre of the face. The convection part can be written as

$$\rho \left(v_i \bar{v} - v_i v_g \right)_{lm} S_{lm} \tag{2.34}$$

and the pressure part as

$$\left(p n_i \right)_{lm} S_{lm} \; . \tag{2.35}$$

The averaged momentum flux $F_{i,lm}$ is then

$$F_{i,lm} = \left[\rho \left(v_i \bar{v} - v_i v_g \right)_{lm} + \left(p n_i \right)_{lm} \right] S_{lm} \tag{2.36}$$

The three velocities in the convection part, i.e. the *convection velocity* $\bar{v}$, the *convected velocity* v_i, and the grid velocity v_g, are all treated separately. The convected component in the flux (2.34) is upwound to the face centre on the basis of the direction of the convection velocity relative to the face $\bar{v} - v_g$. As illustrated in Fig. 2.3, the outer normal $n_{i,lm}$ of a cell face is assumed to point from the control volume l to m. Thus, the convection velocity $\bar{v}_{lm}$ and the grid velocity v_g are positive if the flow or the displacement is from l towards m. The upwinding is taken into account in the convection part of the flux by evaluating the flux as

$$\frac{1}{2}\rho S_{lm}\left[(\bar{v}_{lm} - v_{g,lm})(v^m_{i,lm} + v^l_{i,lm}) - |\bar{v}_{lm} - v_{g,lm}|(v^m_{i,lm} - v^l_{i,lm})\right] , \tag{2.37}$$

where $v^l_{i,lm}$ and $v^m_{i,lm}$ are the extrapolated values of the convected velocity into the face centre lm from the control volumes l and m, respectively. In the following subsections, the details of the discretisation of the convection velocity $\bar{v}_{lm}$ are discussed first. The different approaches adopted for the extrapolation of the convected velocity components $v^{l/m}_{i,lm}$ are discussed later in this section. The discretisation of the grid velocity $v_{g,lm}$ is associated with the temporal discretisation and is thus discussed separately in Sec. 2.4.

The discretisation of the convection velocity

The approximation for the convection velocity is identical to that described by Demirdžić et al. (2000). In orthogonal or nearly orthogonal cases, i.e. when the line connecting neighbouring cell centres is normal to the connecting face and runs through the face centre or deviates only a little from these, the convection velocity $\bar{v}_{lm}$ on the face is calculated as an average of the cell centre values to the left and right of the face. That is

$$\bar{v}^o_{lm} = \left[w_{l,lm} v_{i,l} + (1 - w_{l,lm})\, v_{i,m}\right] n_{i,lm} , \tag{2.38}$$

where $w_{l,lm}$ is a weighting factor depending on the normalised distance of the opposite cell centre m from the face lm evaluated as

$$w_{l,lm} = \frac{\sum_i (x_{i,m} - x_{i,lm})\, n_{i,lm}}{\sum_i (x_{i,m} - x_{i,l})\, n_{i,lm}} . \tag{2.39}$$

In non-orthogonal cases a skewness correction is performed. The convection velocity evaluated using the orthogonal approximation (2.38) is corrected by extrapolation. The correction is based on first-order Taylor extrapolation to the face centre from the intersection of the face and the line connecting the neighbouring cell centres. The idea of the correction is illustrated in Fig. 2.6. The correction can be written as

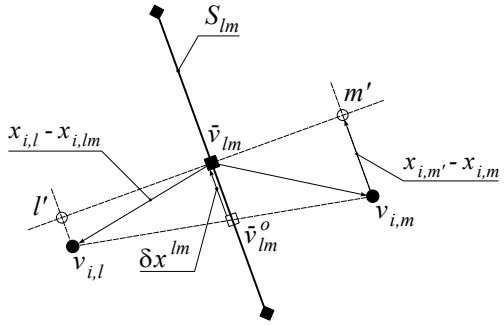


Figure 2.6: In non-orthogonal cases the value $\bar{v}^o_{lm}$ is corrected with skewness correction by extrapolating the value to the face centre (solid square) from the intersection (open square) with first-order Taylor extrapolation. The gradient at the intersection required for the extrapolation is evaluated as a weighted average of the neighbouring cell centre (solid circles) gradients.

$$\delta\bar{v}_{lm} = \delta x_{lm}\frac{\partial v_{i,lm}}{\partial s}n_{i,lm} = \left[w_{l,lm}\delta x^{l'}_{lm} + (1 - w_{l,lm})\,\delta x^{m'}_{lm}\right]\frac{\partial v_{i,lm}}{\partial s}n_{i,lm}\;, \tag{2.40}$$

where the tangential component of the velocity gradient $\partial v_i/\partial s$ on the face lm is approximated with

$$\frac{\partial v_{i,lm}}{\partial s} = \left[w_{l,lm}\frac{\partial v_{i,l}}{\partial x_j} + (1 - w_{l,lm})\,\frac{\partial v_{i,m}}{\partial x_j}\right]s_{j,lm} \tag{2.41}$$

i.e. as a weighted average of the tangential component of the gradients at the neighbouring cell centres. Here, $s_{i,lm}$ is the unit tangent vector of the face. The approximation of the gradients is described in detail in Sec. 2.3.6. The tangential distance δx_{lm} between the intersection of the connecting line and the face centre in Eq. (2.40) can be evaluated in a similar way to the weighted average of the tangential distances of the auxiliary points l' and m' from the cell centres l and m. The auxiliary points are projected from the points l and m, respectively, on the line which is normal to the face lm and runs through the centre point of this face (see Fig. 2.6). Thus, the distances to be averaged can be evaluated as

$$\delta x^{l'}_{lm} = (x_{i,l'} - x_{i,l})\,s_{i,lm} = (x_{i,lm} - x_{i,l})\,s_{i,lm} \tag{2.42}$$

The final approximation for the convection velocity in non-orthogonal cases can be written as

$$\bar{v}_{lm} = \bar{v}^o_{lm} + \delta\bar{v}_{lm}\;. \tag{2.43}$$

The non-orthogonality correction (2.43) is activated for each face separately. The angles between the face normal and the vectors connecting the face centre and the cell centres on both sides of the face are compared with a threshold value. The correction is activated

if either of the angles exceeds the threshold value. However, the correction is not used for faces on the boundary. The same also applies for the rest of the skewness corrections presented below. Skewness corrections on the boundaries would require approximation of the gradients in the ghost cells, which is a non-trivial task. In most cases, disregarding the correction on the boundaries is acceptable. It should be noted that e.g. the correction above is proportional to the distance between the face centre and the intersection of the line $\overrightarrow{lm}$ with the face. Here, $\overrightarrow{lm}$ is the line connecting the cell centres l and m. If the line runs through the face centre, the correction vanishes. As most of the ghost cell centres – except for the boundaries with a symmetry condition applied to them – are created by mirroring the cell centre inside the domain relative to the face centre, the corrections for these faces vanish. It should also be noted that for the same reason the orthogonality of the connecting line is actually not a necessary condition for the zero correction of the convection velocity. However, it is used as a general condition for the activation of the skewness correction in different discretisations. For some of these, orthogonality is a necessary condition for zero correction.

The discretisation of the convected velocity components

For the convected velocity components $v_{i,lm}^{l/m}$ in (2.37) three different discretisations are used in this thesis: a first-order discretisation and two second-order discretisations (Mikkola, 2002, 2003). One of the higher-order schemes is based on weighted Taylor extrapolation and is applicable to arbitrary control volume types. The other one is a 2D variant of the higher -order construction for tetrahedra by Frink (1994) and is only applicable to triangles.

The first higher-order approach is an extension of a very common discretisation based on the first-order Taylor series expansion of the solution within a control volume (see e.g. Anderson, 1992; Murthy and Mathur, 1997; Wang and Liu, 2000). Here, the solution is assumed to vary linearly within the control volume. Thus, the value at an arbitrary point l' within a control volume l can be written as

$$v_i(x_{i,l'}) = v_{i,l} + \frac{\partial v_{i,l}}{\partial x_j}\left(x_{j,l'} - x_{j,l}\right) \tag{2.44}$$

In the method that was developed a modification of this, with weighting of the first-order term, is used. Here the convection velocity on a face is extrapolated as

$$v_{i,lm}^{l} = v_{i,l} + \gamma\frac{\partial v_{i,l}}{\partial x_j}\left(x_{j,lm} - x_{j,l}\right) \tag{2.45}$$

with $0 < \gamma < 1$. This is similar in form to the discretisation introduced by Barth and Jespersen (1989), in which γ is a limiter guaranteeing the monotonicity of the constructed value. However, in the current implementation γ is a global, user-defined constant controlling the overall blending between the first- and second-order upwind discretisations. In this work, only results with $\gamma = 1$ are presented.

The second approach is a modification of a discretisation which relies on the geometrically invariant properties of tetrahedra (Frink, 1994). Triangles are known to have similar invariant properties, so the same kind of approach can be applied to them as well. The construction is based on two known properties. First, it is known that the centre point of a triangle is at the median of the triangle, i.e. on lines connecting a corner node with the centre point of the opposite face. This is illustrated in Fig. 2.7. Second, the distance from

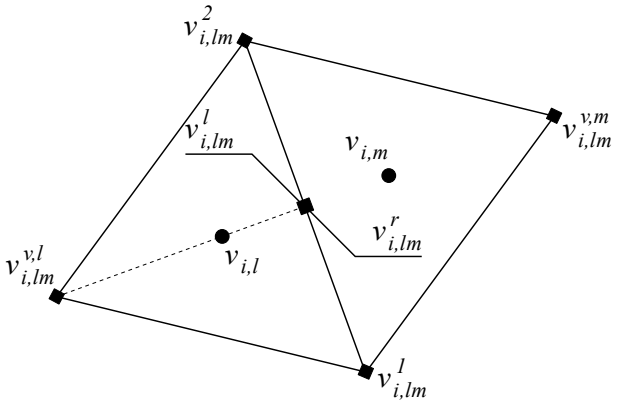


Figure 2.7: If Frink's higher-order extrapolation of the face values is adopted for triangular elements, an auxiliary value at the cell centre (solid square) is evaluated first as the mean of the values at the end nodes of the face (solid diamonds 1 and 2). The change along the median (dashed line) is then evaluated as the difference between the auxiliary value and the value at the opposite node $v^{v,l}_{i,lm}$. The final approximation for the face value is obtained by adding one third of the change to the cell centre (solid circle) value.

the centre point of a face of a triangle to the centre point of the triangle is one third of the length of the corresponding median. The extrapolation is based on the idea that the value at the centre point of a face is first approximated by the mean of the values at the vertices of this face (values $v^{1}_{i,lm}$ and $v^{2}_{i,lm}$ in Fig. 2.7). The change of the variable along a median can now be approximated by subtracting the value at the opposite vertex (values $v^{v,l}_{i,lm}$ and $v^{v,m}_{i,lm}$ in Fig. 2.7) from the previously calculated mean value. It may be assumed that the change of a variable from the centre point of the triangle to the centre point of the face is one third of the change along the median. A final approximation is thus reached

by adding one third of the median change to the cell centre value, or

$$\begin{aligned} v_{i,lm}^{l} &= v_{i,l} + \frac{1}{3}\left(\frac{v_{i,lm}^{1} + v_{i,lm}^{2}}{2} - v_{i,lm}^{v,l}\right) \\ v_{i,lm}^{m} &= v_{i,m} + \frac{1}{3}\left(\frac{v_{i,lm}^{1} + v_{i,lm}^{2}}{2} - v_{i,lm}^{v,m}\right) \end{aligned} \tag{2.46}$$

The evaluation of the values at the vertices will be explained in detail in Section 2.3.5.

As a third alternative a first-order scheme was implemented for the approximation of the convected velocity component. In the case of a first-order approximation, the variable is assumed to be constant within a control volume and thus the face value is simply the value of the variable at the centre of the volume, that is

$$v_{i,lm}^{l} = v_{i,l} \ , \qquad v_{i,lm}^{m} = v_{i,m} \ . \tag{2.47}$$

The approximation (2.47) is also used for the linearisation of the fluxes in the implicit stage, even if a higher-order extrapolation is used in the calculation of the flux terms in the explicit residual. This will be discussed in more detail in Sec. 2.5.1

The discretisation of the pressure in the momentum flux

The pressure term (2.35) in the momentum flux is treated similarly to the convection velocity $\bar{v}_{lm}$. Thus, in orthogonal or nearly orthogonal cases the pressure on the face in Eq. (2.35) is simply taken as a weighted average of the cell centre values on both sides of the face, i.e.

$$p_{lm}^{o} = w_{l,lm} p_l + (1 - w_{l,lm})\, p_m \ . \tag{2.48}$$

The correction in cases of non-orthogonality is identical to the correction (2.40) for the convection velocity. For the pressure it is evaluated as

$$\delta p_{lm} = \delta x_{lm} \frac{\partial p_{lm}}{\partial s} = \left[w_{l,lm} \delta x_{lm}^{l'} + (1 - w_{l,lm})\, \delta x_{lm}^{m'} \right] \frac{\partial p_{lm}}{\partial s} \ , \tag{2.49}$$

where the tangential component of the gradient on the face $\partial p_{lm}/\partial s$ is evaluated similarly to Eq. (2.41). The final approximation for the pressure term of the momentum flux in a non-orthogonal case is

$$(p n_i)_{lm}\, S_{lm} = (p_{lm}^{o} + \delta p_{lm})\, n_{i,lm} S_{lm} \ . \tag{2.50}$$

In addition to the discretisation of the pressure described above, an alternative approach was studied as well (Mikkola, 2007). Using the Gauss theorem, the surface integral of the

product of the pressure and the normal vector in Eq. (2.8) can be expressed as a volume integral of the pressure gradient, i.e.

$$\int_{\partial\Omega} p n_i dS = \int_{\Omega} \frac{\partial p}{\partial x_i} dV \ . \tag{2.51}$$

Now, an approximation for a gradient at a cell centre is readily available as gradients are required for e.g. the construction of the convective flux. Therefore, in the alternative approach the midpoint rule is used to approximate the volume integral, and, thus, the approximation for the pressure gradient at the cell centre is used to evaluate the volume integral. This gives a discrete form

$$\sum_{lm} \left(p n_i\right)_{Slm} S_{lm} = V_l \frac{\partial p_l}{\partial x_i}, \quad lm \in \mathrm{S_l} \tag{2.52}$$

for the influence of the pressure in the momentum equations (2.28).

It should be noted that the two approaches are identical if the Gauss theorem is used for the approximation of the gradients at the cell centres, as then

$$\frac{\partial p_l}{\partial x_i} = \frac{1}{V_l} \sum_{lm} p_{lm} n_{i,lm} S_{lm}, \quad lm \in \mathrm{S_l} \ . \tag{2.53}$$

The approximation of the gradients will be discussed in more detail in Sec. 2.3.6.

2.3.4 The spatial discretisation of the kinematic boundary condition

When the kinematic boundary condition (2.21) is being discretised, approximations have to be constructed for the normal velocity $v_i n_i$ on the free surface, as well as for the time rate of deformation of the surface $\partial h_n / \partial t$. The former is discussed next, whereas the latter is dealt within the context of temporal discretisation in Sec. 2.4.5. The discretisation of the normal velocity on the faces of the free surface boundary appears to be very straightforward, as the discretisation of the convection velocity has already been discussed in the previous section. In the momentum flux the convection velocity is approximated with Eq. (2.38) using the average of the physical and the ghost cell velocities, as well as the outer normal vector of the cell face. The use of this same approximation for the discretisation of the kinematic boundary condition is, however, problematic. The reason for this is not obvious from the form (2.21). Thus, the kinematic boundary condition is first reformulated in order to highlight the characteristics of the condition which have to be taken into account in the discretisation.

The kinematic boundary condition can be written for the different components of the free surface location h_i by combining Eqs. (2.20) and (2.21). This gives

$$\frac{\partial h_i}{\partial t} = v_j n_j n_i \ . \tag{2.54}$$

By writing this out for the two components and by performing some trivial manipulations on the right-hand side, the condition can be written as

$$\begin{aligned}\frac{\partial h_1}{\partial t} &= (v_1 n_1 + v_2 n_2)\, n_1 = v_1 n_1^2 + v_2 n_1 n_2 = v_1 n_1^2 + v_1 n_2^2 - v_1 n_2^2 + v_2 n_1 n_2 \\ &= v_1 + (v_2 n_1 - v_1 n_2)\, n_2 = v_1 + v_t n_2 \end{aligned} \tag{2.55}$$

$$\begin{aligned}\frac{\partial h_2}{\partial t} &= (v_1 n_1 + v_2 n_2)\, n_2 = v_2 n_2^2 + v_1 n_1 n_2 = v_2 n_2^2 + v_2 n_1^2 - v_2 n_1^2 + v_1 n_1 n_2 \\ &= v_2 - (v_2 n_1 - v_1 n_2)\, n_1 = v_2 - v_t n_1 \ , \end{aligned} \tag{2.56}$$

where $v_t = v_2 n_1 - v_1 n_2$ is the tangential component of the velocity at the boundary. In this case, the tangent of the surface is in the direction of the increasing parameter s (see Fig. 2.1). The components of the normal vectors can be given in terms of the free surface derivatives as

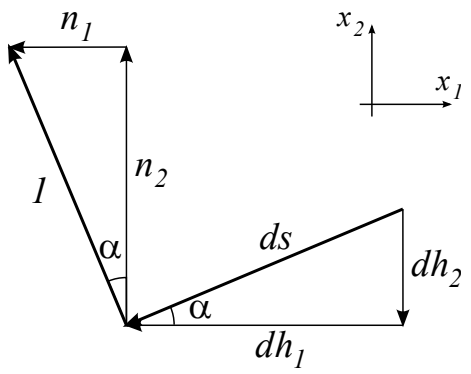


Figure 2.8: The relation between the components of the free surface normal vector and the surface slopes. It should be noted that the parameter s runs from right to left on the free surface (see Fig. 2.1).

$$n_1 = \frac{\partial h_2}{\partial s}, \qquad n_2 = -\frac{\partial h_1}{\partial s} \ . \tag{2.57}$$

This is illustrated in Fig. 2.8. By substituting these into Eqs. (2.55) and (2.56) the two equations take an identical form and the boundary condition becomes

$$\frac{\partial h_i}{\partial t} = v_i - v_t \frac{\partial h_i}{\partial s} \ . \tag{2.58}$$

This reveals that the kinematic boundary condition applied in this work can be cast in a form similar to the familiar kinematic boundary condition for vertical deformations

$$\frac{\partial h}{\partial t} = v_2 - v_1 \frac{\partial h}{\partial x_1} \ . \tag{2.59}$$

This equation consists of a source term and a convection term and has a well-known character. The velocities in the equation do not cause any problems. This implies that the velocity components can be approximated in an identical way to the convection velocity (2.38) as averages of the neighbouring cell centre values, in the kinematic boundary condition (2.21) too. However, the convection term needs special attention.

In the discrete formulation the normal vector $n_{i,lm}$ on a cell face lm is evaluated using the nodes at the ends of the face. Here, it is assumed that on the free surface the nodes are located on the piecewise linear curve connecting the discrete free surface nodes $h_{i,lm}$. This is illustrated in Figs. 2.5 and 2.9. The relation between the free surface nodes and

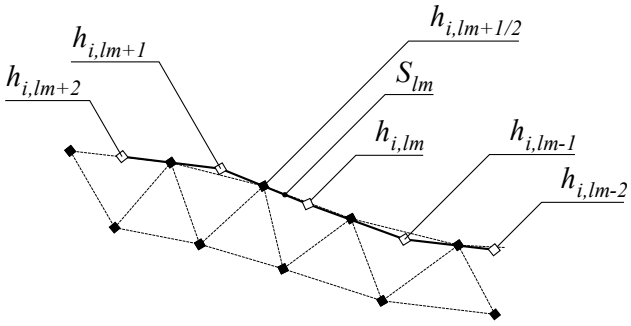


Figure 2.9: The free surface nodes $h_{i,lm}$ and the grid nodes at the free surface boundary. For simplicity of notation the free surface neighbours are denoted by a structured notation, $\pm j$. The actual implementation is, however, unstructured.

the surface normals is (see Eq. 2.57)

$$n_{1,lm} = \frac{h_{2,lm+1} - h_{2,lm-1}}{s_{lm+1} - s_{lm-1}} \qquad n_2 = -\frac{h_{1,lm+1} - h_{1,lm-1}}{s_{lm+1} - s_{lm-1}} \,. \tag{2.60}$$

By substituting this relation into Eqs. (2.55) and (2.56) the discrete condition would be

$$\frac{\partial h_{i,lm}}{\partial t} = v_{i,lm} - v_{t,lm} \frac{h_{i,lm+1} - h_{i,lm-1}}{s_{lm+1} - s_{lm-1}} \,. \tag{2.61}$$

It can be seen that using the normal vectors in the evaluation of the convection velocity in the kinematic boundary condition would be equivalent to using a central difference approximation for the derivatives $\partial h_i / \partial s$ in Eq. (2.58). This in turn would be conjoined with an odd-even decoupling and possible oscillations of the wave heights. In Eq. (2.61) the deformation at the face lm depends only on the neighbouring wave heights, but not on the wave height at the face lm. Thus, any oscillation in the wave height at the face lm would not be captured or suppressed by the equation.

Several authors have suggested that some numerical damping should be included in the

conventional kinematic free surface condition (2.59) by upwinding the slope of the free surface (see e.g. Hino et al., 1993; Raven et al., 2004). Similarly, in this work optional controlled numerical damping is added into the kinematic condition (2.21) through upwinding. This is done by effectively upwinding the free surface slope $\partial h_i/\partial s$ in the convection term of Eq. (2.58) according to the direction of the tangential velocity v_t.

The upwinding can be performed in two alternative ways. In the first alternative the upwinding is performed when the wave height coordinates are being transformed into grid node coordinates. The grid nodes are, thus, evaluated with an upwind scheme and the resulting normal vectors are used in Eq. (2.54). In this case the grid nodes do not generally lie on the free surface definition curve (see Fig. 2.9). In the second alternative the grid nodes always lie on the definition curve, as in Fig. 2.9, but the actual normal vector components $n_{i,lm}$ are not used in the kinematic boundary condition (2.54). Instead, auxiliary normal components are evaluated separately from the wave height coordinates with upwinding. Both alternatives were studied with the method that was developed. In both cases the approximation of the free surface slope is based on a MUSCL-type (Van Leer, 1979) approach.

In (Mikkola, 2004) the author implemented the upwinding in the interpolation of the node coordinates from the neighbouring free surface coordinates. MUSCL interpolation is applied for coordinates normal to the free surface. In the tangential direction grid node locations were taken as a weighted average of the neighbouring free surface coordinates. The slope of the free surface was taken from the normal vector of a free surface face. This approach has proven to be somewhat cumbersome, especially in cases where attempts were made to avoid wave reflection at the boundaries by deliberately adding more numerical damping through lower-order interpolation for the node coordinates.

In order to get better control over the numerical damping and a simpler approach, the author devised an alternative approach in (Mikkola, 2005), which is used in the simulations presented in this thesis. This is based on a similar approach to vertical deformations presented by the author in (Mikkola, 2003). In the method used, instead of the free surface normal vector being used, slope components $\partial h_i/\partial s$ in Eq. (2.57) are calculated separately by a simple MUSCL interpolation of the free surface coordinates. Grid points are always taken as weighted averages of neighbouring wave coordinates.

In the modified implementation of the original MUCSL interpolation applied in this work

the values upwound from the left and right, respectively, are evaluated as

$$\phi^{+}_{lm-1/2} = \phi_{lm} \qquad -\frac{\Delta s_{lm}}{4}\left[(1+k)\,\frac{\Delta\phi_{lm-1/2}}{\Delta s_{lm-1/2}} + (1-k)\,\frac{\Delta\phi_{lm+1/2}}{\Delta s_{lm+1/2}}\right] \tag{2.62}$$

$$\phi^{-}_{lm-1/2} = \phi_{lm-1} + \frac{\Delta s_{lm-1}}{4}\left[(1+k)\,\frac{\Delta\phi_{lm-1/2}}{\Delta s_{lm-1/2}} + (1-k)\,\frac{\Delta\phi_{lm-3/2}}{\Delta s_{lm-3/2}}\right] , \tag{2.63}$$

where k is a parameter allowing switching between different schemes (see Fig. 2.9 for the index notation). A general difference $\Delta\phi_{lm-1/2}$ is defined as

$$\Delta\phi_{lm-1/2} = \phi_{lm} - \phi_{lm-1} . \tag{2.64}$$

If a first-order upwind scheme is used the left- and right-hand values are simply

$$\phi^{+}_{lm-1/2} = \phi_{lm} \qquad \phi^{-}_{lm-1/2} = \phi_{lm-1} . \tag{2.65}$$

Eqs. (2.62) and (2.63) include three differences for a particular variable. These differences include four variable values. For the wave coordinates this means that values on four free surface faces are required. However, the scheme can be simplified by noting that the grid node coordinates are evaluated as weighted averages of the wave coordinates using

$$h_{i,lm-1/2} = \frac{S_{0,lm}h_{i,lm-1} + S_{0,lm-1}h_{i,lm}}{S_{0,lm} + S_{0,lm-1}} , \tag{2.66}$$

where S_0 are the initial free surface face sizes and $h_{i,lm-1/2}$ is used to denote the grid node coordinates. The initial sizes can be used instead of the current face sizes, as the grid nodes are always distributed on the boundaries on the basis of the initial node spacing (see Sec. 2.8). Furthermore, the use of the initial sizes proved to be the more robust of the alternatives. In this case, the evaluation of the distance-related differences Δs in Eqs. (2.62) and (2.63) can also be based on the initial spacing distribution.

In the simplified form of Eqs. (2.62) and (2.63) the difference $\Delta h_{i,lm-1/2}$, which is common to the equations, is evaluated in a normal manner using the wave coordinates and Eq. (2.64) as

$$\Delta h_{lm-1/2} = h_{lm} - h_{lm-1} . \tag{2.67}$$

However, for the other two differences $\Delta h_{i,lm+1/2}$ and $\Delta h_{i,lm-3/2}$ Eq. (2.66) is used and the differences are evaluated as

$$\Delta h_{i,lm+1/2} = \frac{S_{0,lm} + S_{0,lm+1}}{S_{0,lm}}(h_{i,lm+1/2} - h_{i,lm}) \tag{2.68}$$

and similarly for $\Delta h_{i,lm-3/2}$. The differences Δs are evaluated with

$$\Delta s_{lm+1/2} = \frac{1}{2}(S_{0,lm} + S_{0,lm+1}) \,, \tag{2.69}$$

and, therefore, a simple form

$$\frac{\Delta h_{i,lm+1/2}}{\Delta s_{i,lm+1/2}} = \frac{2}{S_{0,lm}}(h_{i,lm+1/2} - h_{i,lm}) \,, \tag{2.70}$$

can be used in Eq. (2.62) and a similar form for $\Delta h_{i,lm-3/2}$ in Eq. (2.63). At the intersection of two boundaries the values upwound onto the outside of the boundary cannot be evaluated with this procedure, unless both boundaries are free surface boundaries, as free surface coordinates on the outside are not available. In this case the coordinates of the intersection are used directly as the values on the outside, whereas the value on the inside can be evaluated with Eq. (2.62) or (2.63). The evaluation of the intersection location will be discussed in Sec. 2.8.

The slope components are evaluated by approximating Eq. (2.57) using either $h^+_{lm\pm1/2}$ or $h^-_{lm\pm1/2}$, depending on the direction of the tangential velocity $v_{t,lm}$, i.e.

$$n^{\pm}_{1,lm} = \frac{h^{\pm}_{2,lm+1/2} - h^{\pm}_{2,lm-1/2}}{S_{lm}} \qquad n^{\pm}_{2,lm} = -\frac{h^{\pm}_{1,lm+1/2} - h^{\pm}_{1,lm-1/2}}{S_{lm}} \,. \tag{2.71}$$

The normal velocity $v_i n_i$ is then approximated as

$$v_{i,lm} n_{i,lm} = \begin{cases} v_{i,lm} n^-_{i,lm}, & \text{if } v_{t,lm} \geq 0 \\ v_{i,lm} n^+_{i,lm}, & \text{if } v_{t,lm} < 0 \,. \end{cases} \tag{2.72}$$

Two fundamental differences result between the approaches to the upwinding. In the former approach the kinematic boundary condition is satisfied exactly and the mass flux through the cell faces on the free surface is zero. In the latter approach the slopes of the cell faces and the approximated free surface slopes in the kinematic boundary condition do not generally match. This results in non-zero mass fluxes (2.26) through the free surface cell faces. On the other hand, in the former alternative the grid nodes defining the boundary of the computational domain do not generally lie on the curve defining the free surface, whereas in the latter alternative they always do.

Both of these differences are caused by the use of upwinding at some point in the process. Because of this, the differences should diminish with grid refinement and vanish at the limit of continuous description. No systematic comparison of the performance of the

two alternatives has been performed, but practical tests have not indicated any significant differences. The latter alternative is preferred for its simplicity of implementation and because the upwinding in that case has a direct influence only on the slopes in the kinematic boundary condition, rather than on the geometry of the domain. Even though the slope calculation in both methods is based on the MUSCL approach, it should be noted that this does not rule out the possibility of using central differencing. By an appropriate choice of the MUSCL parameter ($k = 1$), the scheme reduces to the central difference.

2.3.5 The evaluation of the cell vertex values

For certain discretisations, such as for the Frink discretisation of convected velocities (2.46), the cell vertex values of flow variables are required, in addition to the values at the cell centres and cell face centres. The vertex values are marked with v in the superscript to distinguish them from the cell and face centre values. The value ϕ_q^v at a vertex $q \in \mathrm{N}$ of the grid is approximated by a weighted average (Frink, 1992)

$$\phi_q^v = \frac{\sum_m w_{qm}\phi_m}{\sum_m w_{qm}}, \quad m \in \mathrm{C}_{\mathrm{q,v}}^{\mathrm{nb}} \tag{2.73}$$

of the surrounding cell centre values ϕ_m, where ϕ is one of the state variables, i.e. a component of velocity v_i or pressure p. This is illustrated in Fig. 2.10. In the current method the weight used in the averaging is the inverse of the distance between the vertex and the corresponding cell centre; that is

$$w_{qm} = \left[(x_{1,q}^v - x_{1,m})^2 + (x_{2,q}^v - x_{2,m})^2\right]^{-1/2} . \tag{2.74}$$

For cell vertices lying at the boundary of the computational domain the treatment is slightly different. First, approximations for the values at the centres of the faces on the boundary are calculated using ghost cells lying outside the boundary. After this the vertex values are approximated by a weighted average of the two boundary face values on both sides of the node in a similar way to the average (2.73) above.

2.3.6 The approximation of the gradients

The approximation of a gradient is required for the extrapolation of the components of the convected velocity in Eq. (2.45), as well as for the skewness corrections in Eqs. (2.40)

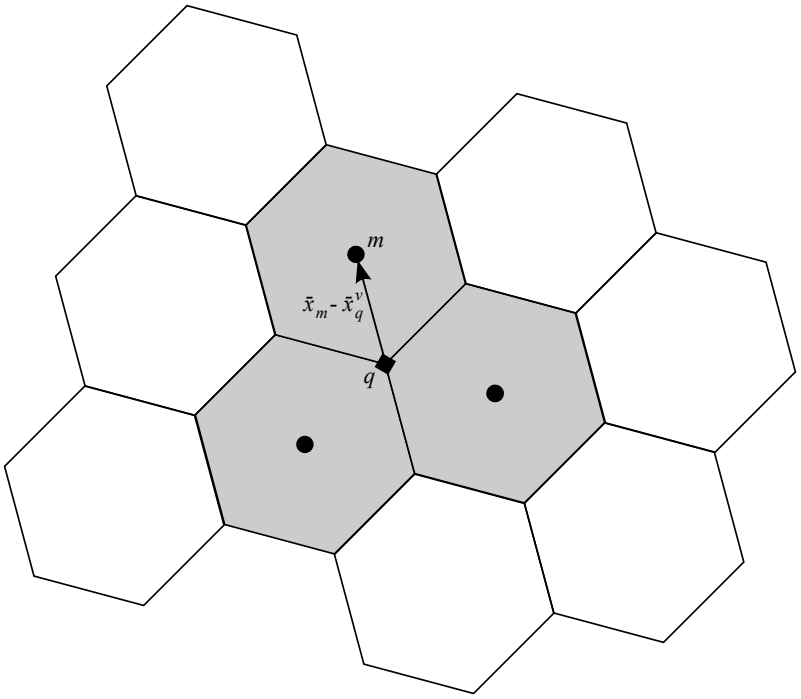


Figure 2.10: Approximation of the value of a flow variable at a cell vertex $q \in \mathrm{N}$ using the values at the centres of the cells surrounding the vertex (shaded). The surrounding cell centres (circles) form the set $\mathrm{C}_{\mathrm{q,v}}^{\mathrm{nb}} \subset \mathrm{C}$.

and (2.49). Furthermore, the gradient approximation is also required if the alternative approach (2.52) is used for the evaluation of the pressure term in the momentum flux.

Two alternative methods are used for the gradient approximation in the method that was developed. The first one (Mikkola, 2003) is based on Gauss integration, whereas in the second one the gradients are constructed using a least-squares method as described by e.g. Demirdžić and Muzaferija (1995). For the sake of completeness, both methods are discussed in detail below.

In the first approach the gradient at a cell centre is approximated using the midpoint rule for a volume integral of the gradient and the Gauss theorem Eq. (2.51), giving

$$\frac{\partial \phi_l}{\partial x_i} = \frac{1}{V_l} \int_{\partial\Omega} \phi n_i dS \tag{2.75}$$

for a general variable ϕ. A midpoint rule is also used for the approximation of the surface integral. The values at the face centres required for the surface integral are evaluated using the skewness-corrected approach identical to Eqs. (2.43) and (2.50). The gradient

is, thus, given by

$$\frac{\partial \phi_l}{\partial x_i} = \frac{1}{V_l} \sum_{lm} \left[w_{l,lm} \phi_{l',lm} + (1 - w_{l,lm}) \, \phi_{m',lm} \right] S_{lm} n_{i,lm}, \quad lm \in \mathrm{S_l} \, , \tag{2.76}$$

where $\phi_{l',lm}$ and $\phi_{m',lm}$ refer to the values at the auxiliary points l' and m', respectively (see Fig. 2.6). The values at the auxiliary points are evaluated from

$$\phi_{l',lm} = \phi_l + \frac{\partial \phi_{lm}}{\partial x_i} \left(x_{i,l'} - x_{i,l} \right) = \phi_l + \frac{\partial \phi_{lm}}{\partial s} \delta x_{lm}^{l'} \tag{2.77}$$

and similarly for the point m'. The approach is identical to the previously presented form of skewness correction, as, instead of the gradient at the corresponding cell centre l or m in Eq. (2.77), the averaged gradient at the face lm is used. By substituting Eq. (2.77) into Eq. (2.76) and by taking into account the fact that the vectors $\overrightarrow{ll'}$ and $\overrightarrow{mm'}$ are tangential to the face, it is easy to see that the approximation consists of an orthogonal approximation and a skewness correction identical to the one in e.g. Eq. (2.50).

Now the gradient at a cell centre appears on both sides of the equation (2.76). Furthermore, the gradient at each cell centre depends on the gradients at the other cell centres through the skewness correction. An iterative approach is used to solve this problem. A user-defined number of evaluations of Eq. (2.76) is performed, taking the gradient components in the skewness correction from the previous evaluation. For the first iteration the gradients are either assumed to be zero or the final values from the previous iteration run are used. Both options have been used in the work. The latter option is preferred as it usually results in a faster convergence of the approximation.

In the second approach to the approximation of the gradients the variation of a variable around a cell centre l is assumed to follow the linear relation

$$\phi(x) = \phi_l + \frac{\partial \phi_l}{\partial x_i} \left(x_i - x_{i,l} \right) \, . \tag{2.78}$$

By writing this for a pair of neighbouring cell centres l and m, an equation with gradient components as unknowns (two in 2D, three in 3D) is produced

$$\frac{\partial \phi_l}{\partial x_i} \left(x_{i,m} - x_{i,l} \right) = \phi_{i,m} - \phi_{i,l} \tag{2.79}$$

or

$$\frac{\partial \phi_l}{\partial x_i} \delta x_{i,\overline{lm}} = \delta \phi_{i,\overline{lm}} \, , \tag{2.80}$$

where the differences are denoted by δ. A system of equations is produced by writing similar equations, using every neighbouring cell centre m of the cell centre l. As the

number of equations is higher than the number of unknowns, the resulting overdetermined system of equations is solved using a least-squares method, i.e. the minimisation problem

$$\min_{\nabla\phi_l} \sum_m \left(\frac{\partial \phi_l}{\partial x_1} \delta x_{1,\overline{lm}} + \frac{\partial \phi_l}{\partial x_2} \delta x_{2,\overline{lm}} - \delta \phi_{\overline{lm}} \right)^2, \quad m \in \mathrm{C_l^{nb}} \tag{2.81}$$

for the sum of the squared residuals is considered. This results in a pair of linear equations

$$\begin{cases} \dfrac{\partial \phi_l}{\partial x_1} \displaystyle\sum_m \delta x_{1,\overline{lm}}^2 + \dfrac{\partial \phi_l}{\partial x_2} \sum_m \delta x_{1,\overline{lm}} \delta x_{2,\overline{lm}} = \sum_m \delta x_{1,\overline{lm}} \delta \phi_{\overline{lm}} \\ \dfrac{\partial \phi_l}{\partial x_2} \displaystyle\sum_m \delta x_{2,\overline{lm}}^2 + \dfrac{\partial \phi_l}{\partial x_1} \sum_m \delta x_{1,\overline{lm}} \delta x_{2,\overline{lm}} = \sum_m \delta x_{2,\overline{lm}} \delta \phi_{\overline{lm}} \end{cases}, \quad m \in \mathrm{C_l^{nb}} \tag{2.82}$$

which is solved trivially using Cramer's rule.

2.4 Temporal discretisation

In the method that was developed time in general – and thus also temporal discretisation – has two fundamentally different meanings. On one hand time refers to actual physical time. In unsteady problems the continuous physical time is discretised into a discrete set of time instants for which the solution is sought. On the other hand time has a non-physical, pseudo-time meaning. In this latter context it is used to march the solution process through intermediate pseudo-time instants towards a final converged solution. Steady-state problems are solved by marching in pseudo-time, starting from some initial state until the solution is sufficiently converged. On the other hand, in unsteady problems each physical time step is considered as a steady-state problem and an identical marching procedure is used, starting from the solution of the previous physical time step.

In addition to the spatial derivatives in the governing equations, a time derivative term appears explicitly in the momentum equations (2.8). In problems with moving boundaries an additional temporal term is included implicitly in the equations, as the grid velocity v_g in Eq. (2.17) is linked to a time derivative through the geometric conservation law (2.11). From the point of view of the temporal discretisation of the governing equations it is sufficient to consider just the momentum equations. The continuity condition (2.13) is, in this regard, uninteresting, as it is simply a condition which a set of values has to satisfy at a single instant of time.

The kinematic boundary condition (2.21) for free surface problems also contains a time derivative. In the following subsections its discretisation is discussed separately. How-

ever, the temporal discretisation and the associated concepts of dual time stepping and pseudo-time discretisation are first discussed in general terms.

2.4.1 The concept of dual time stepping

The momentum equations and the kinematic boundary condition are of the general form

$$\frac{\partial \phi(t)}{\partial t} = R(\phi(t), t) \; , \tag{2.83}$$

where t is the physical time. A corresponding steady-state problem $R = 0$ can be solved by using this same form. However, here the physical time t is replaced with a pseudo-time τ, i.e.

$$\frac{\partial \phi(\tau)}{\partial \tau} = R(\phi(\tau)) \; . \tag{2.84}$$

By starting from some initial guess it is then possible to march in pseudo-time until $\partial\phi/\partial\tau = 0$ and the steady-state equation is satisfied. In practice this means that the equation is integrated over a finite number of pseudo-time steps until the change of the variable is sufficiently small. The pseudo-time differs from the physical time in that it does not have to be constant throughout the domain, i.e. different time steps can be used in different parts of the domain. This stems from the fact that as the interest in steady problems is in the final steady solution, the process of getting there does not have to be physically correct.

The solution of unsteady problems is based on the procedure developed for steady problems. The unsteady problem (2.83) is written as

$$R(\phi(t), t) - \frac{\partial \phi(t)}{\partial t} = 0 \; . \tag{2.85}$$

This is then considered as a steady problem for each time instant and is solved by marching in pseudo-time. Similarly to Eq. (2.84), the equation at the physical time t is thus of the form

$$\frac{\partial \phi(t,\tau)}{\partial \tau} = R(\phi(t,\tau), t) - \frac{\partial \phi(t,\tau)}{\partial t} \; . \tag{2.86}$$

The temporal discretisation deals with two problems: the discretisation of the partial time derivatives $\partial\phi/\partial t$ and $\partial\phi/\partial\tau$ and the approximation of the residual R. As the process for unsteady problems is based on the solution of a steady-state problem, the steady-state discretisation is considered first. The discussion of the implementation of time accuracy is then based on the steady state formulation.

2.4.2 The pseudo-time discretisation

As explained above, the solution process includes the time integration of unsteady equations in pseudo-time. In the method that was developed the time integration is performed using the implicit Euler scheme. For an equation of the form

$$\frac{\partial \phi}{\partial \tau} = R(\phi) \tag{2.87}$$

the implicit Euler scheme reads

$$\frac{\Delta \phi^n}{\Delta \tau} = R(\phi^{n+1}) = R^{n+1} \ , \tag{2.88}$$

where

$$\Delta \phi^n = \phi^{n+1} - \phi^n \tag{2.89}$$

The previous step is denoted by n and the next step by $n+1$. The residual R is now a function of the unknown state variable ϕ on the next step $n+1$. Approximation of the residual by linearising it with respect to the current step n gives

$$R(\phi^{n+1}) \approx R(\phi^n) + \frac{\partial R(\phi^n)}{\partial \phi^n} \Delta \phi^n \ . \tag{2.90}$$

Substituting this into Eq. (2.88) leads to an equation for the change of the state variable ϕ at step n

$$\left(\frac{1}{\Delta \tau} + \frac{\partial R(\phi^n)}{\partial \phi^n} \right) \Delta \phi^n = R^n \ , \tag{2.91}$$

where the right-hand side is the explicit residual. This residual is the "driving force" of the iteration towards a steady state. It is the error made in the steady state equations if the current solution is substituted into them. When the steady-state solution is reached, the error goes to zero and the residual vanishes. As the steady state is determined only by the residual on the right-hand side, it is possible to make rather crude approximations in the evaluation of the derivative on the left-hand side. However, approximations on the left-hand side do affect the convergence properties of the iteration and may lead to a divergent method.

2.4.3 The dual-time stepping and the discretisation of the physical time derivative

The simulation of time-dependent flows is based on a three-level fully implicit scheme (3-LFI) (Hoffren, 1993). As briefly explained above, the method is implemented using dual-time stepping, i.e. two time steps – a physical and a pseudo one – are used (see Eq. 2.86).

The equations applied include both the physical and the pseudo-time derivative terms. Within each physical time step, the problem is considered as a steady-state problem and is iterated until the pseudo-time derivative terms vanish. Thus, for time-accurate cases the starting-point for the discrete form of the equations is the implicit steady-state equation of the form (see Eqs. 2.86 and 2.88)

$$\frac{\Delta\phi(t)^n}{\Delta\tau} = \frac{\phi(t)^{n+1} - \phi(t)^n}{\Delta\tau} = R(\phi(t)^{n+1}, t) - \frac{\partial\phi(t)}{\partial t} \,, \tag{2.92}$$

where n is the pseudo iteration step and t is the physical time. By including an implicit three-level difference approximation for the physical time derivative (Hoffren, 1993) on the right-hand side of this equation, the dual-time step form is given by

$$\frac{\Delta\phi^{n,k+1}}{\Delta\tau} = R^{n+1,k+1} - \frac{3\phi^{k+1} - 4\phi^k + \phi^{k-1}}{2\Delta t} \,, \tag{2.93}$$

where k is the previous physical iteration step corresponding to $t - \Delta t$. The current physical time step, which gives the solution at the physical time t, is denoted by $k + 1$. The unknown solution at the physical iteration $k + 1$ is approximated with the solution at the next pseudo-iteration, i.e $\phi^{k+1} = \phi^{n+1,k+1}$. For the sake of brevity of notation the superscript referring to the current physical time step is hereafter dropped. The pseudo-time steps are always associated with the iteration for the current physical time step and, thus, referring to just the pseudo-time step is sufficient for unambiguous notation. Eq. (2.93) can be written as

$$\frac{\Delta\phi^n}{\Delta\tau} = R^{n+1} - \frac{3\left(\phi^n + \Delta\phi^n\right) - 4\phi^k + \phi^{k-1}}{2\Delta t} \,. \tag{2.94}$$

Here ϕ^{n+1} has been written as $\phi^n + \Delta\phi^n$. Rearranging the terms leads to

$$\left(\frac{3}{2\Delta t} + \frac{1}{\Delta\tau}\right)\Delta\phi^n = R^{n+1} - \frac{3\phi^n - 4\phi^k + \phi^{k-1}}{2\Delta t} \,. \tag{2.95}$$

The linearisation of the residual R^{n+1}, just as with Eq. (2.91), gives the final form

$$\left(\frac{3}{2\Delta t} + \frac{1}{\Delta\tau} - \frac{\partial R(\phi^n)}{\partial\phi^n}\right)\Delta\phi^n = R^n - T^n \,. \tag{2.96}$$

Here

$$T^n = \frac{3\phi^n - 4\phi^k + \phi^{k-1}}{2\Delta t} \tag{2.97}$$

is the explicit part of the approximation for the physical time derivative.

As can be seen from Eq. (2.96), the changes to the steady state equation (2.91) are minor and the implementation of time accuracy in the steady-state version is rather straightforward. It is enough to add a single term dependent on the physical time step into the

diagonal term of the equation system and subtract the current approximation of the physical time derivative from the residual. Of course, in addition to these changes, storage has to be added for the solution at the previous time levels. Furthermore, these values have to be updated after the completion of a physical time step.

2.4.4 The discretisation of the grid velocity

The temporal discretisation of the momentum equations is not limited to the discretisation of the physical and pseudo-time derivatives $\partial/\partial t$ and $\partial/\partial\tau$. Eq. (2.17) contains an additional unsteady term in the form of the normal component of the grid velocity v_g. According to the geometric conservation law (2.12), for constant density the surface integral of the grid velocity has to be equal to the time rate of change of the size of $\Omega(t)$.

The discrete equivalent of the GCL is based on the regions swept by the faces of the cell between the physical time steps (see Fig. 2.11). The discrete GCL written for an arbitrary

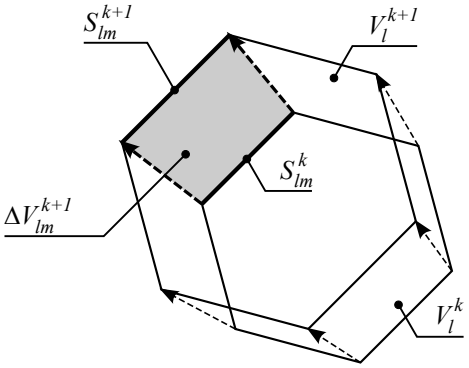


Figure 2.11: The grid velocity $v_{g,lm}$ on a face lm is evaluated on the basis of the size ΔV_{lm}^{k+1} of the region swept by the face between the physical time steps k and $k+1$ (shaded).

polygonal control volume l is

$$\frac{\partial}{\partial t} V_l = \sum_{lm} v_{g,lm} S_{lm}, \quad lm \in \mathrm{S_l}\,. \tag{2.98}$$

By using the 3-LFI for the temporal discretisation of this, it becomes

$$\frac{3V_l^{k+1} - 4V_l^k + V_l^{k-1}}{2\Delta t} = \sum_{lm} v_{g,lm}^{k+1} S_{lm}^{k+1}, \quad lm \in \mathrm{S_l}\,. \tag{2.99}$$

The terms on the left-hand side can be rearranged to give

$$\frac{3\left(V_l^{k+1} - V_l^k\right) - \left(V_l^k - V_l^{k-1}\right)}{2\Delta t} = \sum_{lm} v_{g,lm}^{k+1} S_{lm}^{k+1}, \quad lm \in \mathrm{S_l}\,. \tag{2.100}$$

The changes on the left-hand side are decomposed so as to consist of the regions swept by the faces of the cell l between the physical time levels $k-1$ and k, as well as k and $k+1$. The total change between the time steps can be given as the sum of the sizes of these swept regions. This gives (see Fig. 2.11 for the notations)

$$\sum_{lm} \frac{1}{2\Delta t}\left(3\Delta V_{lm}^{k+1} - \Delta V_{lm}^k\right) = \sum_{lm} v_{g,lm}^{k+1} S_{lm}^{k+1}, \quad lm \in \mathrm{S_l}\,. \tag{2.101}$$

By requiring the equality to be satisfied separately for each term of the sum, the grid velocity can be evaluated as

$$v_{g,lm}^{k+1} = \frac{1}{2S_{lm}^{k+1}\Delta t}\left(3\Delta V_{lm}^{k+1} - \Delta V_{lm}^k\right)\,. \tag{2.102}$$

If the flow problem includes only boundaries with predefined movement, the geometry of the domain and the corresponding grid node locations at any instant in time are known exactly. Thus, the grid velocity can be evaluated directly from Eq. (2.102). However, if the problem involves free surface boundaries, the unknown node locations at the current physical time are approximated explicitly with the known values at the previous pseudo-time step.

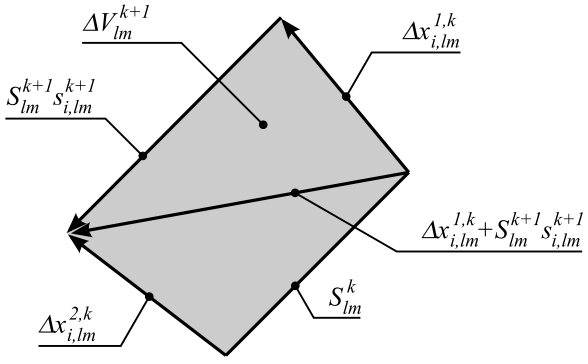


Figure 2.12: The size ΔV_{lm}^{k+1} of the region swept by the face lm between the physical time steps k and $k+1$ is evaluated by dividing the region into two triangles.

The size ΔV_{lm} of the region swept by a face in Eq. (2.102) is evaluated by dividing the region into two triangles (see Fig. 2.12). The areas of the triangles are evaluated using the

cross-product of the vectors forming the triangles. The resulting total size of the swept region is

$$\begin{aligned}\Delta V_{lm}^{k+1} &= \frac{\varepsilon_{3ij}}{2} S_{lm}^{k+1} \Delta x_{i,lm}^{1,k} s_{j,lm}^{k+1} + \frac{\varepsilon_{3ij}}{2} \Delta x_{i,lm}^{2,k} (\Delta x_{j,lm}^{1,k} + S_{lm}^{k+1} s_{j,lm}^{k+1}) \\ &= \frac{S_{lm}^{k+1}}{2} (\Delta x_{i,lm}^{1,k} + \Delta x_{i,lm}^{2,k}) n_{i,lm}^{k+1} + \frac{\varepsilon_{3ij}}{2} \Delta x_{i,lm}^{1,k} \Delta x_{j,lm}^{2,k} \ ,\end{aligned} \tag{2.103}$$

where Δx are the displacements of the nodes at the ends of the face. The cross-products are denoted with the Levi-Civita or permutation symbol ε_{ijk}, which is 1 if the index triplet ijk is an even permutation of $\{1, 2, 3\}$, -1 if it is an odd permutation, and 0 if any index is repeated.

2.4.5 The implicit discretisation of the kinematic boundary condition

The temporal discretisation of the kinematic boundary condition (2.18) is based on the dual-time stepping approach discussed in Sec. 2.4.3. Thus, the condition is expressed as a steady-state problem in pseudo-time, similarly to Eq. (2.86). For the form (2.18) of the condition the approach results in

$$\frac{\partial h_n}{\partial \tau^f} = v_i n_i - v_i^f n_i \ , \tag{2.104}$$

where τ^f is the pseudo-time for the free surface solution. If the form Eq. (2.21) of the kinematic boundary condition is used instead, the condition becomes

$$\frac{\partial h_n}{\partial \tau^f} = v_i n_i - \frac{\partial h_n}{\partial t} \ . \tag{2.105}$$

The pseudo-time discretisation of the kinematic boundary condition is an extension of a scheme for structured grids and vertical deformations first presented by the author in (Mikkola, 2000), which is similar to the approach of Tahara and Stern (1996). It is based on the implicit Euler scheme outlined in Sec. 2.4.2. Thus, the right-hand sides of Eqs. (2.104) and (2.105) are approximated at the future pseudo-time step, i.e.

$$\frac{\Delta h_n^n}{\Delta \tau^f} = R^{f,n+1} - T^{f,n+1} \ , \tag{2.106}$$

where $R^f = v_i n_i$ represents the normal component of the flow velocity on the free surface, and $T^f = v_i^f n_i$ or $T^f = \partial h_n / \partial t$ represents the velocity of the free surface and depends on the form of the kinematic boundary condition used. In the following subsections the approximation and linearisation of these two terms are discussed.

The treatment of the normal component of the flow velocity

The linearisation of the normal velocity $v_i n_i$ on a free surface face lm gives

$$(v_{i,lm}n_{i,lm})^{n+1} = v^n_{i,lm}n^n_{i,lm} + v^n_{i,lm}\sum_j \frac{\partial n^n_{i,lm}}{\partial h^n_{n,lm+j}}\Delta h^n_{n,lm+j}, \quad lm+j \in \mathrm{B_f}\ , \tag{2.107}$$

where $\mathrm{B_f}$ is the union of the free surface boundary sections. The scheme is not fully implicit in pseudo-time, as the velocity components are not linearised, but is approximated by the values in the previous pseudo-iteration.

For the evaluation of the partial derivatives in the last term on the right-hand side of Eq. (2.107), the change of the normal vector as a function of the free surface deformation is considered. This is illustrated in Fig. 2.13. As the length of the normal vector is

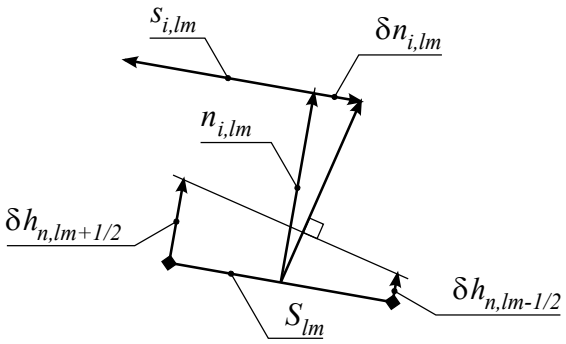


Figure 2.13: The change of the face normal vector with the deformation of the free surface. The free surface deformations and the resulting rotation of the normal vector are assumed to be infinitesimal.

constant, the infinitesimal change of the vector δn_i has to be perpendicular to the normal vector n_i and, thus, parallel to the tangent vector s_i. The change of the normal vector can be given in terms of the change in the free surface slope as (see Fig. 2.13)

$$\delta n^n_{i,lm} = -\frac{\delta h^n_{n,lm+1/2} - \delta h^n_{n,lm-1/2}}{S_{lm}} s^n_{i,lm}\ . \tag{2.108}$$

For the linearisation the slope terms are approximated with the first-order upwind scheme in order to maintain a compact stencil. In order to simplify the scheme it is assumed that the directions of the normal vectors do not change significantly over the three neighbouring faces that make a non-zero contribution to the sum in Eq. (2.107). In Eq. (2.108) these

approximations mean that $\delta h_{n,lm+1/2}$ and $\delta h_{n,lm-1/2}$ are given with

$$\delta h_{n,lm+1/2} = \begin{cases} \delta h_{n,lm}, & \text{if } v_{t,lm} \geq 0 \\ \delta h_{n,lm+1}, & \text{if } v_{t,lm} < 0 \end{cases} \tag{2.109}$$

$$\delta h_{n,lm-1/2} = \begin{cases} \delta h_{n,lm-1}, & \text{if } v_{t,lm} \geq 0 \\ \delta h_{n,lm}, & \text{if } v_{t,lm} < 0 \, . \end{cases} \tag{2.110}$$

By combining these with Eq. (2.108) the differentials in the sum of Eq. (2.107) can be written as

$$\begin{cases} \dfrac{\partial n_{i,lm}}{\partial h_{n,lm}} \Delta h_{n,lm} &= -\dfrac{v_{t,lm}}{|v_{t,lm}| S_{lm}} s_{i,lm} \Delta h_{n,lm} \\ \dfrac{\partial n_{i,lm}}{\partial h_{n,lm-1}} \Delta h_{n,lm-1} &= \dfrac{\max(v_{t,lm}, 0)}{|v_{t,lm}| S_{lm}} s_{i,lm} \Delta h_{n,lm-1} \\ \dfrac{\partial n_{i,lm}}{\partial h_{n,lm+1}} \Delta h_{n,lm+1} &= \dfrac{\min(v_{t,lm}, 0)}{|v_{t,lm}| S_{lm}} s_{i,lm} \Delta h_{n,lm+1} \\ \dfrac{\partial n_{i,lm}}{\partial h_{n,lm+j}} \Delta h_{n,lm+j} &= 0, \quad \text{if } j \notin \{-1, 0, 1\} \, . \end{cases} \tag{2.111}$$

The substitution of these relations into Eq. (2.107) and the fact that $v_i s_i = v_t$ give the final, linearised approximation of the normal velocity in the future pseudo-time step

$$\begin{aligned} (v_{i,lm} n_{i,lm})^{n+1} = v^n_{i,lm} n^n_{i,lm} &- a^{f,n}_{lm} \Delta h^n_{n,lm} \\ &- a^{f,n}_{lm-1} \Delta h^n_{n,lm-1} - a^{f,n}_{lm+1} \Delta h^n_{n,lm+1} \, , \end{aligned} \tag{2.112}$$

where

$$\begin{cases} a^{f,n}_{lm} &= -a^{f,n}_{lm-1} - a^{f,n}_{lm+1} \\ a^{f,n}_{lm-1} &= -\dfrac{\max(v^n_{t,lm}, 0)}{S^n_{lm}} \\ a^{f,n}_{lm+1} &= \dfrac{\min(v^n_{t,lm}, 0)}{S^n_{lm}} \, . \end{cases} \tag{2.113}$$

The Δh_n terms are transferred to the left-hand side of Eq. (2.106) to give

$$\begin{aligned} \left(\frac{1}{\Delta \tau^f_{lm}} + a^{f,n}_{lm} \right) \Delta h^n_{n,lm} + a^{f,n}_{lm-1} \Delta h^n_{n,lm-1} &\\ + a^{f,n}_{lm+1} \Delta h^n_{n,lm+1} = R^{f,n}_{lm} + T^{f,n+1}_{lm} \, . & \end{aligned} \tag{2.114}$$

The left-hand side of this equation forms the steady part of the linear system. In steady problems $T^f \equiv 0$ and Eq. (2.114) is used to solve the free surface deformations Δh_n.

Time accuracy with direct approximation of the time derivative

In the time-accurate solution of free surface flows the second term $T^{f,n+1}$ on the right-hand side of Eq. (2.106) has to be considered as well. This consists of the approximation and linearisation of the term representing the normal component of the velocity of the free surface. On the basis of Eq. (2.104), the intuitive choice for the approximation of the free surface velocity would be to use the grid velocity in the kinematic boundary condition. However, it has been observed by the author (Mikkola, 2005) that this may lead to oscillatory solutions. Therefore, an alternative approximation has been used in this work. Here, the temporal derivative of the wave height in Eq. (2.105) has been approximated with an implicit three- level scheme. This is briefly discussed next.

The treatment of the physical time derivative is identical to the one presented in Sec. 2.4.3. An implicit, three-level approximation is therefore used for the derivative to give

$$\frac{\Delta h^n_{n,lm}}{\Delta \tau^f_{lm}} = R^{f,n+1}_{lm} - \frac{1}{2\Delta t}(3h^{n+1}_{n,lm} - 4h^k_{n,lm} + h^{k-1}_{n,lm}) \; , \tag{2.115}$$

with the solution at the future physical time approximated with the value in the next pseudo-iteration $n+1$. The linearisation of this form is trivial, as h^{n+1}_n can be written as $h^n_n + \Delta h^n_n$. By rearranging the terms, the final form is

$$\begin{aligned}\left(\frac{1}{\Delta \tau^f_{lm}} + \frac{3}{2\Delta t} + a^{f,n}_{lm}\right)\Delta h^n_{n,lm} + \left(a^{f,n}_{lm-1} + \frac{3}{8\Delta t}\right)\Delta h^n_{n,lm-1} \\ + \left(a^{f,n}_{lm+1} + \frac{3}{8\Delta t}\right)\Delta h^n_{n,lm+1} = R^{f,n}_{lm} + T^{f,n}_{lm} \; ,\end{aligned} \tag{2.116}$$

where the coefficients a^f are given in Eq. (2.113).

2.5 The bulk flow solution method

In the previous sections a detailed description was given of both the spatial and temporal discretisation of the governing equations, i.e. of the construction of the discrete form of the governing equations. However, so far very little has been said about the process used to solve the equations.

The bulk flow solution process is based on a SIMPLE- (Caretto et al., 1972) or SIMPLEC-type (van Doormal and Raithby, 1984) velocity-pressure decoupling (see Fig. 2.14). Thus,

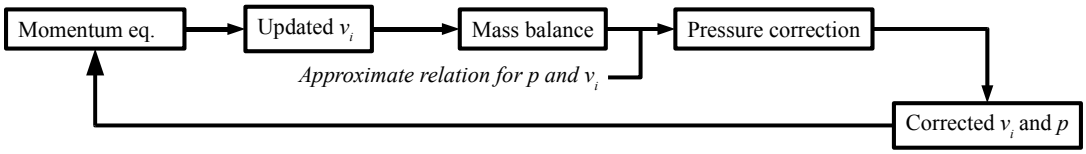


Figure 2.14: The bulk flow solution algorithm is based on velocity-pressure decoupling. Here, the velocity field resulting from the momentum equations with specified pressure is corrected by adjusting the pressure field on the basis of the mass imbalance from the continuity equation.

the velocities and pressures are solved separately in an iterative manner. In each iteration, the velocity field is first updated onto the next pseudo-time step τ using the momentum balance equations (2.28) with the current approximation for the pressure field. After this the provisional velocity field resulting from Eq. (2.28) is corrected by altering the approximation for the pressure field according to the resulting mass balance error in the continuity equation (2.25). The corrected velocity field satisfies the mass balance, but may – and mostly will – violate the momentum balance. Thus, the process is repeated in pseudo-time until both balances are satisfied – in practice, to a specified accuracy. A detailed description of the implementation of each step is given in the following sections.

2.5.1 Momentum equations

The first step in each iteration is the calculation of the change of the velocity components in each control volume from equations with the form (2.91) or (2.96), where now the residual R consists of the sum of the momentum fluxes F_m on the cell faces. The explicit residual on the right-hand side of Eq. (2.28) is easily calculated by summing the fluxes (2.36) over the sides of each volume using the current velocity and pressure. For the implicit stage, however, derivatives of the fluxes on the left-hand side are required (see Eq. 2.91). Each velocity component is treated separately and for the calculation of the change of one component the other component is assumed to be constant. Furthermore, the pressure is treated separately from the velocities and it is also assumed to be constant at this point. Thus, the discrete flux $F_{i,lm}$ can be considered to be a function of the cell centre values of the current velocity component only.

In Eq. (2.90) the flux was assumed to be a function of just a single variable and thus the derivative was taken with respect to this variable only. In the discrete form, however,

a flux can, in principle, be a function of all the cell centre values. Thus, for the total differential, the derivatives have to be taken with respect to all these values and the partial differentials have to be summed. Then the momentum equations for the control volume l can be written as

$$\frac{\rho V_l}{\Delta\tau}\Delta v_{i,l}^n + \sum_p a_{lp}^n \Delta v_{i,p}^n = R_{i,l}^n, \qquad p \in \mathrm{C} \tag{2.117}$$

$$\left(\frac{\rho V_l}{\Delta\tau} + \frac{3\rho V_l}{2\Delta t}\right)\Delta v_{i,l}^n + \sum_p a_{lp}^n \Delta v_{i,p}^n = R_{i,l}^n - T_{i,l}^n, \quad p \in \mathrm{C} \tag{2.118}$$

for the steady and unsteady approaches, respectively. Here a_{lp} is the sum of the derivatives of the fluxes with respect to the cell centre value $v_{i,p}$ over the set of faces $\mathrm{S_l}$ of volume l. That is

$$a_{lp} = \sum_{lm} \frac{\partial F_{i,lm}}{\partial v_{i,p}}, \quad lm \in \mathrm{S_l} \tag{2.119}$$

and

$$R_{i,l} = -\sum_{lm} F_{i,lm}, \quad lm \in \mathrm{S_l} \tag{2.120}$$

is the explicit residual. $T_{i,l}^n$ in the unsteady case is the explicit approximation (2.97) for the physical time derivative. For the sake of brevity, the superscripts in Eqs. (2.119) and (2.120) have been dropped.

As mentioned in Sec. 2.4.2, it is possible to use different approximations for the fluxes in the explicit and implicit stages. The approximation (2.45) or (2.46) used in the calculation of the explicit residuals would result in large stencils and a complex equation system in the implicit stage. Therefore the simpler approximation discussed in Sec. 2.3.3 is used and the convected velocity component is approximated with first-order accuracy by Eq. (2.47). Furthermore, Picard linearisation (Ferziger and Perić, 1997) is used for the linearisation of the momentum flux. Here, the convection velocity $\bar{v}_{lm}$ is assumed to be independent of the cell centre velocities $v_{i,l}$ and is thus left out of the differentiation. This assumption can again be justified by the fact that the solution is determined by the right-hand side of Eq. (2.117) or (2.118) and even crude approximations can be made on the left-hand side. In the case of a steady problem (2.117) or an unsteady problem (2.118) with a fixed grid these approximations give

$$a_{ll} = \sum_{lm} \rho S_{lm} \max\left(\bar{v}_{lm}, 0\right), \quad lm \in \mathrm{S_l} \tag{2.121}$$

for the diagonal element and

$$\begin{cases} a_{lp} = -\rho S_{lp} \max\left(-\bar{v}_{lp}, 0\right), & \text{if } p \in \mathrm{C_l^{nb}} \\ a_{lp} = 0, & \text{if } p \notin \mathrm{C_l^{nb}} \end{cases} \tag{2.122}$$

for the off-diagonal elements. In the case of an unsteady problem (2.118) with grid deformation the corresponding diagonal and off-diagonal elements are

$$a_{ll} = \sum_{lm} \rho S_{lm} \max\left(\bar{v}_{lm} - v_{g,lm}, 0\right), \quad lm \in \mathrm{S_l} \tag{2.123}$$

and

$$\begin{cases} a_{lp} = -\rho S_{lp} \max\left(-\bar{v}_{lp} + v_{g,lp}, 0\right), & \text{if } p \in \mathrm{C_l^{nb}} \\ a_{lp} = 0, & \text{if } p \notin \mathrm{C_l^{nb}} \end{cases} . \tag{2.124}$$

respectively.

A further approximation is made by assuming that the velocities appearing in the coefficients of the momentum equations (2.121) and (2.122) or (2.123) and (2.124) satisfy the continuity equation. The discrete forms of the continuity conditions (2.3) and (2.10) can be expressed as

$$\sum_{lm} \rho S_{lm} \max\left(\bar{v}_{lm}, 0\right) = \sum_{lm} \rho S_{lm} \max\left(-\bar{v}_{lm}, 0\right), \quad lm \in \mathrm{S_l} \tag{2.125}$$

and

$$\begin{aligned} &\sum_{lm} \rho S_{lm} \max\left(\bar{v}_{lm} - v_{g,lm}, 0\right) \\ =&\sum_{lm} \rho S_{lm} \max\left(-\bar{v}_{lm} + v_{g,lm}, 0\right) - \frac{\partial \rho V_l}{\partial t}, \quad lm \in \mathrm{S_l} \end{aligned} \tag{2.126}$$

for the fixed and deforming grid cases, respectively. In Eqs. (2.125) and (2.126) the sums on the left- and right-hand sides are the outflow from and inflow into the cell l, respectively. By comparing the terms in Eqs. (2.121) and (2.122) or (2.123) and (2.124) with the left- and right-hand sides of Eq. (2.125) or (2.126) it can be seen that the diagonal term A_l can be written as

$$A_l = \frac{\rho V_l}{\Delta\tau} + a_{ll} = \frac{\rho V_l}{\Delta\tau} - \sum_{m} a_{lm}, \qquad m \in \mathrm{C_l^{nb}} \tag{2.127}$$

$$A_l = \frac{\rho V_l}{\Delta\tau} + \frac{3\rho V_l}{2\Delta t} + a_{ll} = \frac{\rho V_l}{\Delta\tau} + \frac{3\rho V_l}{2\Delta t} - \sum_{m} a_{lm}, \quad m \in \mathrm{C_l^{nb}} \tag{2.128}$$

for steady and unsteady cases, respectively. In the case of an unsteady problem with deforming geometry the last term on the right-hand side of Eq. (2.126) is neglected. This can be justified quite easily. The purpose of Eqs. (2.127) and (2.128) is to ensure that the resulting linear system is diagonally dominant, even when the continuity condition is not satisfied. A positive time derivative of the control volume size, i.e. an increase in the cell size, would reduce the diagonal dominance of the system and a negative derivative, even

if it increased the diagonal dominance, would be negligible in comparison to the second term on the right-hand side of Eq. (2.128).

Eq. (2.117) or (2.118) with the appropriate diagonal and off-diagonal terms forms the system of equations for the change of the velocity components. The equation system is solved first for the v_1 component, after which the treatment is repeated for the v_2 component. The choice of the pseudo-time step in the time integration is based on the Courant-Friedrichs-Lewy (CFL) condition. The time step is evaluated from a user-defined Courant number $C_{\Delta\tau}$, which is defined as

$$C_{\Delta\tau} = \max_m \left(\frac{\bar{v}_{lm}}{\delta x_{i,\overline{lm}} n_{i,lm}} \right) \Delta\tau_l, \; m \in \mathrm{C_l^{nb}} \; . \tag{2.129}$$

Either Gauss-Seidel or CGSTAB (Conjugate Gradient Stabilized, van den Vorst and Sonneveld, 1990) with D-ILU preconditioning (Diagonal Incomplete Lower Upper, Barrett et al., 1993) is used for the solution. The default choice is CGSTAB. With the changes of both components being known, the velocities are updated.

2.5.2 Pressure correction equation

The next step in the segregated solution process depicted in Fig. 2.14 is to check whether the updated velocity field resulting from the momentum equation step satisfies the continuity condition. It is likely that this is not the case, as continuity is not enforced during the solution of the momentum equations described above. In pressure correction methods this provisional velocity field is corrected by adjusting the pressure in the domain. This affects the velocity through the momentum equations.

The unknown flow variables satisfying the continuity equation can be given as

$$v_i = v_i^* + v_i' \qquad p = p^* + p' \; , \tag{2.130}$$

where v_i^* are the provisional velocity components from the momentum equations, $p^* = p^n$ is the pressure from the previous pressure correction step, and v_i' and p' are the unknown corrections. By substituting the velocities in Eq. (2.130) into the continuity equation (2.25) an equation for the mass flux corrections

$$\sum_{lm} \dot{m}'_{lm} = -\sum_{lm} \dot{m}^*_{lm}, \quad lm \in \mathrm{S_l} \tag{2.131}$$

is produced, where

$$\dot{m}'_{lm} = \rho \bar{v}'_{lm} S_{lm} \; , \tag{2.132}$$

and $\bar{v}'_{lm}$ is the correction for the convection velocity on the face lm.

The pressure correction equation is based on the simple relation (2.131). An equation for the pressure corrections is reached by giving the changes in mass flux on the left-hand side in terms of changes in pressure. The right-hand side depends on the known convection velocities and represents the error in mass balance after the solution of the momentum equations. The convection velocity and the associated discretisation were already discussed within the context of the convection flux (see Eqs. 2.34 and 2.43). However, the approximation of the convection velocity $\bar{v}^*_{lm}$ on the right-hand side of Eq. (2.131) differs from the discretisation used with the momentum equations. Thus, two separate aspects of the construction of the pressure correction equation have to be discussed, i.e. the approximation of the relation between the changes in velocities and pressures and the discretisation of the convection velocity on the basis of the provisional velocity components.

Coupling of the pressure and velocity corrections

The connection between the change in the pressure and the change in the convection velocity $\bar{v}$ on a face can be derived from the momentum equations. Substituting the as yet unknown values (2.130) for velocity components and pressure into Eq. (2.117) or (2.118), and taking into account the fact that the provisional values v^*_i and p^* satisfy the momentum equations, leads to

$$A_l v'_{i,l} + \sum_p a_{lp} v'_{i,p} = - \sum_{lp} S_{lp} p'_{lp} n_{i,lp} \; , \quad p \in \mathrm{C_l^{nb}}, \; lp \in \mathrm{S_l} \; . \tag{2.133}$$

It should be noted that this same form applies in both a steady and an unsteady case. The only difference is in the definition of the diagonal term A_l, which is given with Eq. (2.127) or (2.128). For consistency with the momentum equation discretisation, the pressure correction on a face is defined as

$$p'_{lp} = w_{l,lp} p'_l + (1 - w_{l,lp}) \, p'_p \; . \tag{2.134}$$

Eq. (2.133) gives the relation between the velocity and pressure changes in a cell and its neighbours. However, the form is too complex and, thus, several simplifications of it

have been presented in the literature. The situation with the pressure correction equation is similar to the one with the momentum equations: the left-hand side of Eq. (2.131) can be modified without affecting the steady-state solution as the driving force for the velocity corrections is the mass imbalance on the right-hand side.

In the method that was developed two very common approaches to the approximation of the relation (2.133) were used. In the SIMPLE method (Caretto et al., 1972) the term including the velocity corrections in the neighbouring cells on the left-hand side of Eq. (2.133) is simply neglected. This results in a simple relation

$$A_l v'_{i,l} = -\sum_{lp} S_{lp} p'_{lp} n_{i,lp}, \quad lp \in \mathrm{S_l} \,. \tag{2.135}$$

This is a rather inconsistent approximation, as the neglected term is of the same order of magnitude as the remaining term on the left-hand side. In the alternative SIMPLEC (SIMPLE, consistent) method (van Doormal and Raithby, 1984; Ferziger and Perić, 1997) a more consistent approximation is made by first adding and subtracting $\sum_p a_{lp} v'_{i,l}$ on the left-hand side of Eq. (2.133) and omitting the term

$$\sum_p a_{lp} \left(v'_{i,p} - v'_{i,l} \right), \quad lp \in \mathrm{S_l} \,. \tag{2.136}$$

This is equal to the approximation of the velocity correction at the centre of the control volume l with a weighted average of the neighbouring velocity corrections, in which the matrix multipliers a_{lp} of the neighbours are used as weights. This gives

$$v'_{i,l} = \frac{\sum_p a_{lp} v'_{i,p}}{\sum_p a_{lp}}, \quad p \in \mathrm{C_l^{nb}} \,, \tag{2.137}$$

which, by rearranging the terms, can be written as

$$\sum_p a_{lp} v'_{i,p} = \sum_p a_{lp} v'_{i,l}, \quad p \in \mathrm{C_l^{nb}} \,. \tag{2.138}$$

If this is substituted into Eq. (2.133), the relation between the velocity and the pressure corrections becomes

$$\left(A_l + \sum_p a_{lp} \right) v'_{i,l} = -\sum_{lp} S_{lp} p'_{lp} n_{i,lp} \,, \quad p \in \mathrm{C_l^{nb}}, \; lp \in \mathrm{S_l} \,. \tag{2.139}$$

The relation can be simplified even further using the definitions of the diagonal terms (2.127) and (2.128). These give

$$\frac{\rho V_l}{\Delta \tau} v'_{i,l} = -\sum_{lp} S_{lp} p'_{lp} n_{i,lp} \,, \quad lp \in \mathrm{S_l} \tag{2.140}$$

$$\rho V_l \left(\frac{1}{\Delta \tau} + \frac{3}{2\Delta t} \right) v'_{i,l} = -\sum_{lp} S_{lp} p'_{lp} n_{i,lp} \,, \quad lp \in \mathrm{S_l} \tag{2.141}$$

for the steady and unsteady cases, respectively. This particular simplification is a special case and results from the fact that continuity has been assumed in the derivation of Eqs. (2.127) and (2.128).

In order to have a unified presentation for the SIMPLE and SIMPLEC a general form

$$A_l^p v'_{i,l} = -\sum_{lp} S_{lp} p'_{lp} n_{i,lp}, \quad lp \in \mathrm{S_l}\,, \tag{2.142}$$

of the relation is used. With SIMPLE (see Eq. 2.135)

$$A_l^p = A_l \tag{2.143}$$

and with SIMPLEC (see Eqs. 2.140 and 2.141)

$$\begin{cases} A_l^p = \dfrac{\rho V_l}{\Delta\tau}, & \text{if steady} \\ A_l^p = \rho V_l \left(\dfrac{1}{\Delta\tau} + \dfrac{3}{2\Delta t} \right), & \text{if unsteady.} \end{cases} \tag{2.144}$$

Eq. (2.142) gives the relation between the corrections of the Cartesian velocity components and the pressure. The relation for the convection velocity corrections is obtained by taking the dot product of (2.142) and the normal vector on a face. Hence,

$$A_l^p \bar{v}'_{l\cdot lm} = -\sum_{lp} S_{lp} p'_{lp} \bar{n}_{lp\cdot lm}, \quad lp, lm \in \mathrm{S_l} \tag{2.145}$$

gives an approximation for the change in the velocity component at the centre of cell l in the direction of the normal of the face lm and

$$\bar{n}_{lp\cdot lm} = n_{i,lp} n_{i,lm} \tag{2.146}$$

is the dot product between the unit normals of the face lp and lm. Using the Gauss theorem, the right-hand side of Eq. (2.145) can be written as a finite volume approximation for a volume integral and therefore

$$A_l^p \bar{v}'_{l\cdot lm} = -V_l \frac{\partial p'_l}{\partial x_i} n_{i,lm}\,. \tag{2.147}$$

The relation for the convection velocity $\bar{v}_{lm}$ on the face lm is taken by averaging the relations for the cells l and m. Eq. (2.147) written for the convection velocity correction on the face lm is

$$\bar{v}'_{lm} = -\frac{V_{lm}}{A_{lm}^p} \left(\frac{\partial p'}{\partial n} \right)_{lm}\,. \tag{2.148}$$

Taking V_{lm} and A^p_{lm} as the means of the respective values in the cells l and m and the normal derivative of the pressure correction as the difference in the values at the cell centres on both sides of the face divided by the distance of the centres normal to the face, the final relation is

$$\bar{v}'_{lm} = -\frac{V_l + V_m}{A^p_l + A^p_m} \frac{p'_m - p'_l}{(x_{j,m} - x_{j,l})\, n_{j,lm}} \,. \tag{2.149}$$

Substituting this into Eq. (2.131) leads to the pressure correction equation

$$\alpha_{ll} p'_l + \sum_m \alpha_{lm} p'_m = -\sum_{lm} \dot{m}^*_{lm}, \quad m \in \mathrm{C}^{\mathrm{nb}}_{\mathrm{l}},\ lm \in \mathrm{S}_{\mathrm{l}}, \tag{2.150}$$

where the diagonal elements are

$$\alpha_{ll} = -\sum_m \alpha_{lm}, \quad m \in \mathrm{C}^{\mathrm{nb}}_{\mathrm{l}} \tag{2.151}$$

and the off-diagonal elements are

$$\alpha_{lm} = -\frac{\rho S_{lm}}{(x_{j,m} - x_{j,l})\, n_{j,lm}} \frac{V_l + V_m}{A^p_l + A^p_m} \,. \tag{2.152}$$

The approximation for the normal derivative of pressure used above in Eq. (2.149) is adequate, assuming that the cell centres l and m to the left and right of the face lm lie almost orthogonally to the face and that the line connecting the cell centres intersects the face close to the centre point of the face. However, if this is not the case, the error is taken into account iteratively by using the deferred correction approach (Ferziger and Perić, 1997). For this, a more accurate approximation for the normal derivative is used. The approach used is very similar to the skewness correction discussed in Sec. 2.3.3. The normal derivative on the face lm is calculated similarly as above, but using the values at the auxiliary points l' and m' (see Fig. 2.6). These are determined using the first-order Taylor approximation. Thus, for e.g. l', the value at the auxiliary point is given by

$$p'_{l'} = p'_l + \frac{\partial p'_l}{\partial x_i} (x_{i,l'} - x_{i,l}) \,. \tag{2.153}$$

In the first step the primary pressure correction p' is solved as before, using the cell centre values in the evaluation of the normal derivative and, thus, neglecting the first-order terms in the Taylor approximation (2.153). In the second step a deferred correction method first suggested by Khosla and Rubin (1974) is used. Thus the corrected pressure correction p'' is determined by using

$$\frac{\partial p''}{\partial n} = \left(\frac{\partial p''}{\partial n}\right)^{low} + \left[\left(\frac{\partial p'}{\partial n}\right)^{high} - \left(\frac{\partial p'}{\partial n}\right)^{low}\right] \tag{2.154}$$

for the normal derivative in Eq. (2.148). The superscripts *low* and *high* are used for approximations neglecting and including the first-order terms in Eq. (2.153), respectively. The idea in Eq. (2.154) is that the error made using the lower-order approximation is corrected by the difference between the higher- and lower-order approximations in the previous step. By writing out the approximations in Eq. (2.154) it can be cast into the form

$$\left(\frac{\partial p''}{\partial n}\right)_{lm} = \frac{1}{(x_{j,m}-x_{j,l})n_{j,lm}}\left\{ (p''_m - p''_l) + \right. \\ \left. + \left[\frac{\partial p'_m}{\partial x_i}(x_{i,m'}-x_{i,m}) - \frac{\partial p'_l}{\partial x_i}(x_{i,l'}-x_{i,l})\right]\right\} . \qquad (2.155)$$

Combining Eqs. (2.131), (2.148), now for v'' in terms of p'', and (2.155), a second pressure correction equation is obtained. The first-order term in Eq. (2.155) can be calculated explicitly as the values for p' are known and the term is therefore moved to the right-hand side of the equation. The resulting equation

$$\sum_m \alpha_{lm} p''_m = \\ - \sum_m \left\{ \dot{m}^*_{lm} + \alpha_{lm}\left[\frac{\partial p'_m}{\partial x_i}(x_{i,m'}-x_{i,m}) - \frac{\partial p'_l}{\partial x_i}(x_{i,l'}-x_{i,l})\right]\right\} \qquad (2.156)$$

has the same left-hand side as the first pressure correction equation (2.150) and only the right-hand side has to be updated with the explicit part of the deferred correction (2.154).

Mass flux

Mass fluxes through the sides of the control volumes are required for the calculation of the mass imbalance in the continuity equation driving the pressure correction. In practice this means that an approximation for the convection velocity $\bar{v}^*_{lm}$ on a face lm in Eq. (2.131) has to be known. For the momentum equations the convection velocity on a face is calculated according to Eq. (2.43) as a skewness-corrected average of the cell centre values on both sides of the face. It is tempting to use this same approach here as well, but this may result in a checkerboard pressure field as a result of the weakening of the coupling between neighbouring velocities and pressures. The source of this problem is well known and it is related to the use of a collocated variable arrangement with central approximations both for the convection velocity in the mass balance error and for the pressure in the momentum flux. The matter is discussed extensively in the literature (see e.g. Ferziger and Perić, 1997) and will not be covered further here.

A common approach used for the calculation of the mass flux in the continuity equation with a collocated variable arrangement is the velocity interpolation scheme of Rhie and Chow (1983). In this approach, the difference between two alternative pressure gradient approximations is used to adjust the convection velocity interpolation on a face. Modifications of the scheme for unstructured grids exist as well, e.g. by Davidson (1996) and Thomadakis and Leschziner (1996). An identical approach was used in the method that was developed.

Effectively, these lead to fourth-order damping similar to that commonly used with central differencing to stabilise the solution of the momentum equations Jameson et al. (1981). Considering that the underlying purpose of the method is to add artificial damping into the solution process in order to prevent pressure oscillations, the modification of the Rhie-Chow approach for unstructured grid geometries may lead to an overly complicated scheme. Therefore, an alternative approach using a simplified pressure-damping term and applicable to meshes composed of triangles was implemented in the method as well (for details see e.g. Mikkola, 2002, 2003).

With both approaches the mass flux on the face lm can be written as

$$\dot{m}^*_{lm} = \rho S_{lm} \left(\bar{v}^*_{lm} + C \frac{V_{lm}}{A^p_{lm}} d_{lm} \right) , \tag{2.157}$$

where C is a user-defined parameter. V_{lm} and A^p_{lm} are the averages of the control volume sizes V and diagonal terms A^p at the cell centres l and m, respectively. The convection velocity $\bar{v}^*_{lm}$ is evaluated using Eq. (2.43). The only difference between the approaches lies in the definition of d_{lm} and, thus, the practical implementation of different approaches in the method is a very straightforward task.

In Rhie-Chow-type approaches, such as the one used in this work, the problem with possible checkerboard solutions is circumvented by explicitly introducing the connection between neighbouring pressures into the pressure correction equation through the convection velocity interpolation. First, the approximate influence of the problematic pressure gradient discretisation used for the momentum flux is subtracted from the convection velocity. Then the influence of an alternative pressure gradient discretisation retaining the connection between neighbours is added. The influence of the pressure on the convection velocity is approximated in a similar way to Eq. (2.148). Using Eq. (2.157), this approach leads to a term with the form

$$d_{lm} = \overline{\left(\frac{\partial p}{\partial n} \right)}_{lm} - \left(\frac{\partial p}{\partial n} \right)_{lm} , \tag{2.158}$$

where the first derivative, denoted by an overline, is evaluated as an average of the derivatives at the neighbouring cell centres l and m, and the second derivative is an approximation constructed directly on the face. The evaluation of the first derivative has to correspond to the discretisation of the uncorrected convection velocity $\bar{v}^*_{lm}$ in Eq. (2.157). Thus, the derivative is evaluated as a weighted average of the cell centre values. For the sake of simplicity, no skewness correction is applied, even if it would be applied for the convection velocity. For the second term the normal derivative of the pressure is discretised in an identical way to the normal derivative of the pressure corrections in Eq. (2.149). These give

$$d_{lm} = w_{l,lm}\frac{\partial p_l}{\partial n} + (1 - w_{l,lm})\frac{\partial p_m}{\partial n} - \frac{p_{m'} - p_{l'}}{(x_{i,m'} - x_{i,l'})\, n_{i,lm}} , \tag{2.159}$$

where $p_{l'}$ and $p_{m'}$ refer to the pressures at the auxiliary points l' and m', respectively (see Fig. 2.6).

For the traditional Rhie-Chow-type approach the parameter C in Eq. (2.157) should be 1. However, if the term (2.158) is considered to be a damping term, it is obvious that other values for the parameter can also be used. It should be noted that as the term is simply a difference between two discrete approximations for the normal component of the pressure gradient on the face, it is strictly related to numerics. In the continuous framework the term (2.158) vanishes. Similarly, the term is directly proportional to the grid spacing and, thus, it decreases with grid refinement.

The solution of the pressure correction equation

The pressure correction equation is solved with the Gauss-Seidel, CGSTAB, or CG (Conjugate Gradient, Golub and van Loan, 1990) methods. For the latter two D-ILU preconditioning is implemented. The default choice for the solution is CG.

After the pressure corrections have been solved, the velocities and pressures are updated. However, practice has shown that for a stable calculation, pressure corrections may have to be under-relaxed at this point. This is a known feature of the SIMPLE algorithm (Patankar, 1980). Pressures are updated as

$$p_l^{n+1} = p_l^n + \alpha_p p'_l , \tag{2.160}$$

where p' is replaced by p'' if deferred correction is used. The parameter α_p is the under-relaxation factor for the pressure corrections and with SIMPLE a value between 0 and

1 should be given, depending on the case. However, with SIMPLEC under-relaxation is generally unnecessary and $\alpha_p = 1$ can be used. The velocity components v_i are updated using

$$v_{i,l}^{n+1} = v_{i,l}^{*} + \alpha_v v_{i,l}' \,, \tag{2.161}$$

where the corrections $v_{i,l}'$ are calculated from Eq. (2.142) using either p' or p''. The factor α_v allows for the relaxation of the velocity components as well, in order to improve stability in some cases. In the simulations presented here, however, there was no need for relaxation and a value of $\alpha_v = 1$ was used.

2.5.3 The algorithm

Now that the individual parts of the bulk flow solution have been discussed, the algorithm can be summarised with the following sequence of steps. The steps are presented for a case with a fixed geometry. The extension of the algorithm for problems with a deforming grid is discussed separately in Secs. 2.6 and 2.8.

1. Initialise the velocities and pressures in the computational domain – in unsteady problems also for the old time steps k and $k-1$.
2. Start a new physical time step and update the time $t \rightarrow t + \Delta t$.
3. Start a new pseudo-time step.
4. Update the ghost cell values according to the boundary conditions.
5. Calculate the fluxes for v_1 from Eq. (2.36).
6. Calculate the explicit residuals of v_1 from Eq. (2.120) or Eqs. (2.97) and (2.120).
7. Determine the matrix coefficients (2.122) and (2.127) or (2.122) and (2.128) for the implicit stage.
8. Solve the change of v_1 from the linear system (2.117) or (2.118).
9. Repeat steps 5–8 for the velocity component v_2.
10. Update the velocity components.
11. Update the ghost cell values according to the boundary conditions.

12. Calculate the mass imbalance on the right-hand side of Eq. (2.150) using mass fluxes (2.157).

13. Determine the coefficients (2.151) and (2.152) of the pressure correction equation.

14. Solve the pressure corrections from (2.150).

15. In the case of two pressure corrections, calculate the explicit part of the deferred correction (2.154) using Eq. (2.155) and add it to the mass imbalance.

16. Solve the corrected pressure corrections from Eq. (2.156).

17. Update the pressures according to Eq. (2.160) with under-relaxation.

18. Calculate the velocity corrections from Eq. (2.142) and apply the corrections according to Eq. (2.161).

19. Repeat steps 3–18 until convergence is reached.

20. Update the values on the old physical time steps: $\phi^{k-1} = \phi^k$, $\phi^k = \phi^{k+1}$.

21. Return to step 2.

A steady-state algorithm is restricted to the initialisation of the problem and the inner iteration, which consists of the steps 3–19

2.6 Coupling of the free surface and bulk flow solutions

A common approach to solving free surface problems with surface tracking is to decouple the bulk flow and free surface problems (see the discussion in Chapter 1). In that case, the solution of the flow consists of two separate steps iterated in turns until a converged solution is reached (see Fig. 2.15). For each iteration, in the bulk flow step the flow field is updated with a free surface resulting from the previous free surface step. In the free surface step the new free surface location is solved on the basis of the updated bulk flow and the grid is adjusted to match the new boundary. The calculation of the new wave height can, in principle, be based on either the kinematic or the dynamic boundary condition. The next bulk flow step is then based on the boundary conditions on the updated free surface.

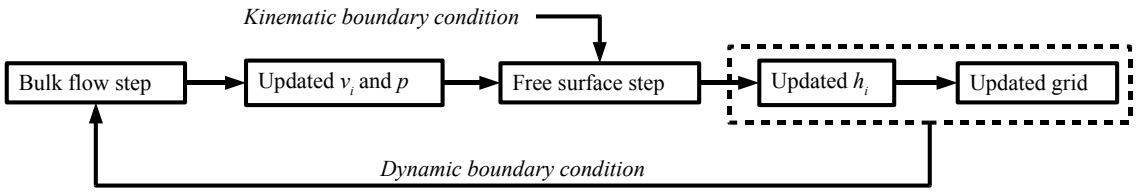


Figure 2.15: Traditional, surface tracking approaches for free surface flows are based on the full decoupling of the bulk flow and free surface solutions, which are iterated in turn. In the example depicted the surface deformation is evaluated with the kinematic boundary condition. The bulk flow solution provides an updated velocity field for the free surface step, which in turn provides an updated free surface location for the grid updating and bulk flow solution processes.

With full decoupling of the solutions one of the free surface boundary conditions – generally the kinematic boundary condition – is chosen for the evaluation of the surface deformation in the free surface step. The other free surface boundary condition is then used on the instantaneous free surface in the bulk flow step. By the definition of full decoupling, both boundary conditions cannot be enforced simultaneously in the bulk flow step, as one of these is used for the evaluation of the deformation. The problem with this is that boundary conditions enforced separately are guaranteed to be compatible only when the iteration has converged, but not during the iteration. This is assumed to have an impact on the stability and convergence of the approach.

The author has developed two alternative methods (Mikkola, 2003) in which the approach presented above is improved by partially coupling the bulk flow and free surface solutions through the pressure correction equation. These exploit the segregated nature of the pressure correction approach. In order to avoid the complication with incompatible boundary conditions, the deforming free surface is taken into account during the pressure correction stage. The approaches are named after the boundary condition used to resolve the surface deformations. In the kinematic approach the free surface deformation is evaluated with the kinematic boundary condition and the dynamic boundary condition is used to take the surface deformation into account in the pressure correction stage. The dynamic approach is based on the evaluation of the free surface deformation from the dynamic boundary condition and implicit relaxation of the zero mass flux condition on the free surface in the pressure correction stage.

2.6.1 The kinematic approach

In the kinematic approach the updating of the wave height is based on the kinematic boundary condition. Full decoupling of the bulk flow and free surface solutions would result in a constant pressure boundary condition on the free surface in the bulk flow step, implying zero pressure correction on the free surface. On the other hand, the deformation of the free surface in the free surface step combined with the dynamic boundary condition changes the pressure on the free surface. This incompatibility of boundary conditions would lead to a jump in pressure across the free surface during the iteration, affecting the overall convergence and stability of the method. In the method that was developed this is alleviated by taking the surface deformation into account with the dynamic boundary condition during the solution of the pressure correction equation. A general illustration of the steps of the process is presented in Fig. 2.16. A more detailed description of the

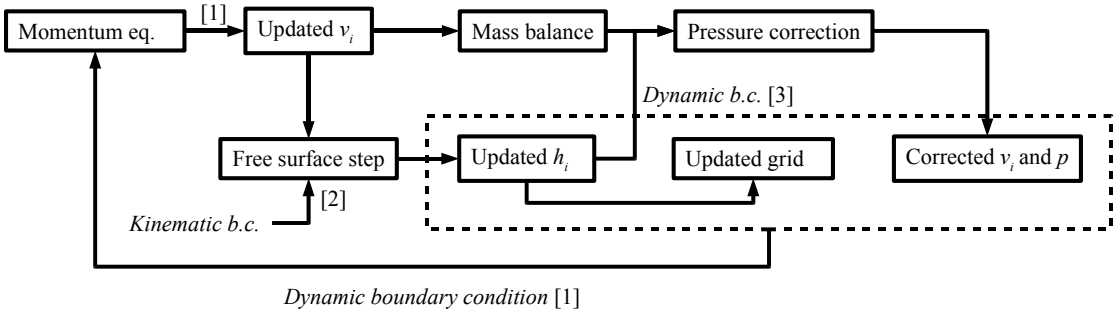


Figure 2.16: The coupling of the bulk flow and free surface solutions with the kinematic approach exploits the segregated nature of the pressure correction method. The free surface deformation is evaluated from the kinematic boundary condition using the velocity field resulting from the solution of the momentum equations. The pressure change resulting from the deformation is then used as a boundary condition for the pressure correction equation.

steps of the process is given next.

For the solution of the governing equations on the pseudo-iteration $n+1$ the grid is assumed to deform with the velocity v_g^n and to coincide with the instantaneous free surface at the pseudo-iteration n. In the case of a steady problem the grid velocity $v_g = 0$. The velocity components are updated from the momentum balance (2.117) or (2.118) using the dynamic boundary condition (2.24) for the pressure and either the zero or first-order extrapolation for the velocity components on the free surface ([1] in Fig. 2.16). Because the velocity is extrapolated from the domain, the kinematic boundary condition is not

necessarily satisfied by the flow at this point.

After the calculation of the mass balance error, the change in the wave height is evaluated by integrating the kinematic boundary condition (2.21) in pseudo-time with an implicit Euler scheme [2], as discussed in detail in Sec. 2.4.5. A simple Gauss-Seidel scheme is used for the solution of the linear system of equations (2.116). The free surface pseudo-time step is evaluated as

$$\Delta\tau^f_{lm} = C_{\Delta\tau^f} \min\left(\frac{S_{lm}}{v_{t,lm}}, \Delta\tau_l, \Delta t\right) \ , \quad lm \in \mathrm{B_f} \ , \tag{2.162}$$

where $C_{\Delta\tau^f}$ is a user defined coefficient.

The pressure on the new free surface has to satisfy the dynamic boundary condition [3], giving the relation

$$p'_{lm} = \rho g \Delta h_{2,lm} \ , \quad lm \in \mathrm{B_f} \tag{2.163}$$

between the pressure correction on the free surface and the change in the wave height. For the solution of the pressure correction equation (2.150), linear extrapolation is used for the ghost cell values, so that Eq. (2.163) is satisfied exactly on the free surface, i.e.

$$p'_m = 2\rho g \Delta h_{2,lm} - p'_l \ , \quad lm \in \mathrm{B_f} \ . \tag{2.164}$$

Because of the relaxation of the pressure corrections, wave heights have to be under-relaxed by the same amount in order to get a free surface compatible with the corrected pressure field. The wave heights are therefore updated as

$$h^{n+1}_{i,lm} = h^n_{i,lm} + \alpha_p \Delta h^n_{n,lm} n^n_{i,lm} \ , \quad lm \in \mathrm{B_f} \ . \tag{2.165}$$

The approach ensures compatible pressures on the free surface at each stage of the iteration. Furthermore, as the pressure change resulting from the surface deformation is taken into account in the pressure correction equation as a boundary condition, the influence of the change in surface pressure is propagated immediately into the whole domain.

2.6.2 The dynamic approach

In the second approach used in the method that was developed, the calculation of the new wave height is based on the dynamic boundary condition. With full decoupling of

the solutions the kinematic boundary condition would then be used in the bulk flow step, enforcing zero mass flux through the free surface. As the dynamic boundary condition is not enforced during the fully decoupled bulk flow step, the pressure can develop on the free surface independently of the surface deformation. Experience has shown that this would lead to large pressure changes and large deformations of the free surface, making the method very unstable, unless considerable under-relaxation of the pressures is applied after the solution of the pressure correction equation. Heavy under-relaxation would, however, have a dramatic effect on the convergence speed of the solution process. In the dynamic approach the process is improved by taking the deformation of the free surface into account in the pressure correction stage. This is done by implicitly relaxing the zero mass flux condition on the free surface. The steps of the process are shown in Fig. 2.17 and are next explained in more detail.

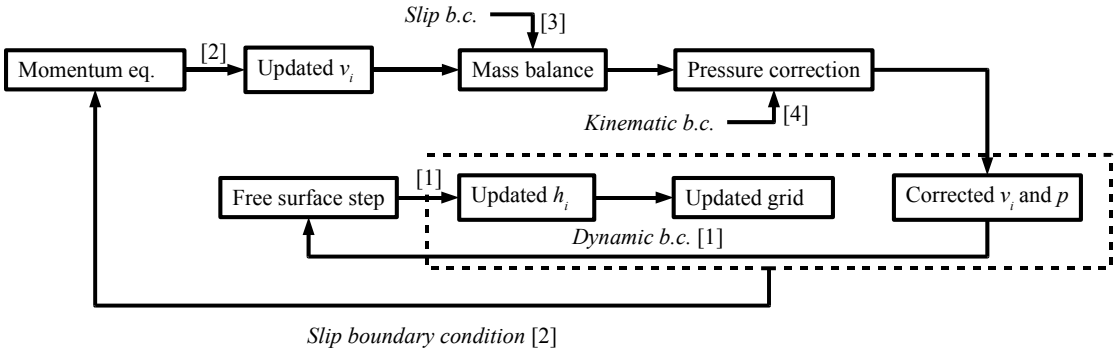


Figure 2.17: In the dynamic approach the free surface deformation is evaluated from the dynamic boundary condition using the pressure field resulting from the solution of the pressure correction equation. The coupling of the bulk flow and free surface solutions under-relaxes the pressure correction equation through the kinematic boundary condition. In the solution of the momentum equations the kinematic boundary condition forces the flow to follow the free surface shape.

Each iteration starts with the calculation of the current wave height from the pressure on the surface ([1] in Fig. 2.17) using the dynamic boundary condition (2.24):

$$h_{2,lm} = \frac{p_{lm}}{\rho g} , \quad lm \in \mathrm{B_f} . \tag{2.166}$$

As with the kinematic approach, the grid is assumed to match the instantaneous free surface resulting from the previous pseudo-iteration n and to deform with the grid velocity v_g^n. Time integration of the momentum equations is done with the kinematic boundary condition used on the free surface [2]. As this enforces zero mass flux on the free surface

and leaves the pressure unspecified, the boundary condition is treated by applying the slip condition to the free surface. The practical implementation of the slip condition is discussed in more detail in Sec 2.7. This same condition is used for the calculation of the mass balance error as well [3].

In this approach, the partial coupling of the bulk flow and free surface solutions is in the form of implicit under-relaxation in the pressure correction stage based on the deforming free surface. As in the first method, the pressure correction on the free surface and the change in wave height can be related through the dynamic boundary condition giving

$$\Delta h_{2,lm} = \frac{p'_{lm}}{\rho g} , \quad lm \in \mathrm{B_f} . \tag{2.167}$$

If the grid is assumed to deform with a grid velocity v_g^n, corresponding to the previous free surface location, this additional deformation of the free surface related to the change in pressure results in a mass flux through the face of the grid given by the kinematic boundary condition as

$$\dot{m}^*_{lm} = \frac{f_l p'_l + (1 - f_l)\, p'_m}{g \Delta \tau^f_{lm}} S_{lm} n_{2,lm} , \quad lm \in \mathrm{B_f} . \tag{2.168}$$

Here l is the computational cell with a face lm on the free surface and m the associated ghost cell. This mass flux is substituted [4] into the right-hand side of the pressure correction equation (2.150). Using zero-order extrapolation for the pressure corrections in Eq. (2.168), i.e. $p'_m = p'_l$ and transferring the contribution to the left-hand side, the diagonal term of the computational cell l with a face lm on the free surface can be written as

$$\alpha^f_{ll} = \alpha_{ll} + \frac{S_{lm} n_{2,lm}}{g \Delta \tau^f_{lm}} , \quad lm \in \mathrm{B_f} . \tag{2.169}$$

The modified pressure correction equation is then solved to give the pressure and velocity corrections. The velocities and pressures are then updated in the normal fashion using Eq. (2.160).

It is easy to see from Eq. (2.169) that the approach increases the weight of the diagonal term. Thus, the approach can be considered to function through implicit under-relaxation of the pressure corrections. This is assumed to reduce the resulting pressure corrections and the corresponding surface deformations and, as a result, improve the stability of the approach compared to a fully decoupled approach based on the unmodified pressure correction equation.

2.7 Discretisation of the boundary conditions

The discussion above on the spatial discretisation in the method that was developed was general in the sense that each face was assumed to have neighbours on both sides of the face. In this sense faces at the boundaries of the domain are an exception, as they have physical neighbours only on one side of the face. The values of these neighbours are required for the evaluation of the fluxes in both the momentum and continuity equations. Furthermore, the values for the neighbouring cells are also required on the right-hand sides of the implicit momentum and pressure correction equations. In the method that was developed this problem was circumvented by using ghost cells, as explained in Sec. 2.3. Boundary faces are also exceptional in another sense. Some of the flow quantities on the boundaries are specified by the boundary conditions, while some of them are determined as part of the flow solution. In the following subsections the discretisation of the boundary conditions is discussed from two points of view. On one hand, the boundary conditions are considered from the point of view of whether the condition specifies a value at the boundary or not. On the other hand, both the explicit and the implicit treatments of the boundary conditions are discussed.

2.7.1 The explicit treatment

The values of the flow quantities in the ghost cells are set in such a way that the flux terms evaluated on a boundary face satisfy the boundary conditions exactly. This applies to boundaries, on which the boundary condition specifies the values of the flow variables on the boundary. The fluxes on the rest of the boundaries become specified as part of the solution based on the conservation property of the finite volume method and on the discretisation of the relevant boundary conditions. In order to demonstrate the correctness of the fluxes, the resulting mass flux and the convection part of the momentum flux are shown in the following for boundary conditions, which specify fluxes through the boundary. The resulting values are then compared with the values corresponding to the boundary condition.

If a boundary condition specifies the value of a variable on the face (Dirichlet boundary condition), the value in the corresponding ghost cell centre is extrapolated linearly using the specified value on the face and the known cell centre value within the domain. Thus,

$$\phi_m = 2\phi_{lm} - \phi_l \ , \tag{2.170}$$

where ϕ_m, ϕ_{lm} and ϕ_l are the values of a general variable in the ghost cell, at the face and at the real cell centre, respectively. In order to reproduce the specified value at the boundary exactly in flux terms, they have to be evaluated as averages. For convection velocity and pressure this means that the uniform approach without skewness correction (2.38) can be used at the boundaries as well. However, for the convected component the user's choice of discretisation is overridden at the boundaries and replaced with the same central approximation.

On the other hand, some boundary conditions specify the value of the normal derivative at the boundary (Neumann boundary condition). In this case the central difference approximation of the derivative on the face is used to determine the value in the ghost cell. Similarly to the Dirichlet boundary condition, no skewness correction is used for the approximation at the boundary. Furthermore, in this thesis only homogeneous Neumann boundary conditions are used. This simplifies the approximation even further, as it is sufficient to copy the value of the variable from the physical cell centre to the ghost cell one, i.e.

$$\phi_m = \phi_l \ . \tag{2.171}$$

The influence of the omission of the skewness correction with Neumann-type conditions is minimised by generating the ghost cells in this case orthogonally to the face, as explained in Sec. 2.3.

If the boundary condition leaves some variables unspecified, the ghost cell values of the variable for such boundaries are determined through extrapolation from the computational domain. Two approaches to the extrapolation have been used in this work. The first is a general, Taylor extrapolation-based approach and the second is a special approach applicable to grids composed of triangles. The former is identical to the extrapolation in Eq. (2.44) and, thus, the value in the ghost cell is given by

$$\phi_m = \phi_l + \frac{\partial \phi_l}{\partial x_j} \left(x_{j,m} - x_{j,l} \right) \ . \tag{2.172}$$

The latter approach is based on the use of the cell centre, as well as the node-averaged values in the extrapolation. The approach is only applicable to triangles, as it uses known properties of triangles. It is assumed that the node opposite to the face and inside the domain lies on the same line as the physical cell centre and the ghost cell centre. Furthermore, it is assumed that the distance between the node and the physical cell centre is equal to the distance between the cell centres. Both are valid assumptions if grids composed of triangles and the normal ghost cells specified in Sec. 2.3 are used (see Fig. 2.18). Linear

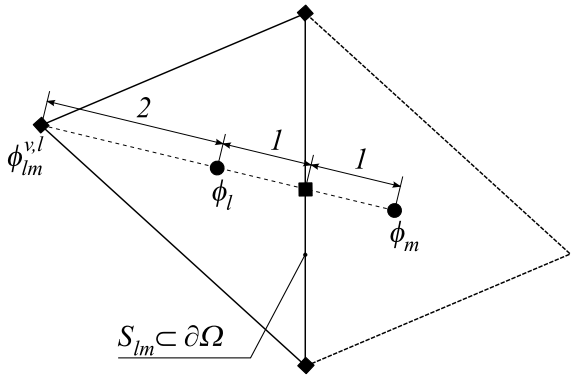


Figure 2.18: The values used for the opposite node-based boundary extrapolation.

extrapolation along the connecting line gives

$$\phi_m = 2\phi_l - \phi_{lm}^{v,l} \,. \tag{2.173}$$

In some cases the boundary condition does not specify the solver unknowns directly, but specifies the value of some derived quantity. One example of such a case is the slip boundary condition, which specifies conditions separately for the normal and tangential components of the velocity at the boundary. In these cases one of the approaches described above is used for the quantity derived. Then the resulting equation for the ghost cell value of the quantity derived is reformulated to use the unknowns of the solver. The process for the slip condition is described in more detail below.

In this work four different boundary condition types have been used. These are the external or free-stream, symmetry, slip, and free surface conditions. The external boundary condition is used at boundaries which are considered to lie sufficiently far away from the source of disturbance in the flow field. The actual type of the condition depends on whether the flow through the boundary is inflow or outflow. In the case of inflow, the velocity components on the boundary are assumed to be fixed. Hence the values in a ghost cell are extrapolated linearly using Eq. (2.170). In order to avoid over-specified boundary conditions, the pressure values are extrapolated from the computational domain using either Eq. (2.172) or (2.173). However, if the flow is directed out of the domain, the pressure is assumed to be fixed and the velocities are extrapolated from the domain. As the convection velocity $\bar{v}_{lm}$ on the boundary is evaluated without the skewness correction

using Eq. (2.38), the mass flux on an inflow type boundary becomes

$$\dot{m}_{lm} = \rho S_{lm} \bar{v}_{lm} = \frac{\rho S_{lm}}{2} \left(v_{i,l} + 2v_{i,lm} - v_{i,l} \right) n_{i,lm} = \rho S_{lm} v_{i,lm} n_{i,lm} \, , \tag{2.174}$$

which is equal to the specified value of the mass flux on the boundary with the midpoint rule. It should be noted that on the boundaries the pressure damping term d_{lm} in the mass flux (2.157) is neglected. The convected velocity components $v_{i,lm}$ are evaluated on the boundaries also as averages. Furthermore, inflow type boundaries are assumed to be fixed, i.e. the grid velocity on the boundary is zero. Thus, the convection part of the momentum flux (2.34) on an inflow type boundary results similarly into

$$\rho \left(v_i \bar{v} - v_i v_g \right)_{lm} S_{lm} = \rho v_{i,lm} v_{j,lm} n_{j,lm} S_{lm} \, . \tag{2.175}$$

This is equal to the specified value of the convection part of momentum flux evaluated using the midpoint rule.

The symmetry boundary condition is used at boundaries where the flow is assumed to be symmetric relative to the boundary. For this kind of boundary, the homogeneous Neumann boundary condition is used for the scalar quantities i.e. in this work for the pressure. Thus, according to Eq. (2.171), the value from the physical cell centre is copied to the ghost cell centre. For velocity the same condition is used for the component tangential to the boundary, which is also copied to the ghost cell. However, as the normal component of the velocity has to be mirrored as well, the normal component in the ghost cell has to be in the opposite direction. This is accomplished by first transferring the velocity into a coordinate system parallel to the boundary, applying the conditions in this coordinate system, and finally transferring back to the original coordinate system. If the required matrix operations are combined into a single operation, the condition can be written in component form as

$$\begin{aligned} v_{1,m} &= \left(1 - 2n_{1,lm}^2 \right) v_{1,l} - 2n_{1,lm} n_{2,lm} v_{2,l} \\ v_{2,m} &= \left(1 - 2n_{2,lm}^2 \right) v_{2,l} - 2n_{1,lm} n_{2,lm} v_{1,l} \, . \end{aligned} \tag{2.176}$$

This discretisation results in zero mass flux, because

$$\begin{aligned} \dot{m}_{lm} = \rho S_{lm} \bar{v}_{lm} = \frac{\rho S_{lm}}{2} \Big[& v_{1,l} n_{1,lm} + \left(n_{1,lm} - 2n_{1,lm}^3 \right) v_{1,l} - 2n_{1,lm}^2 n_{2,lm} v_{2,l} \\ & + v_{2,l} n_{2,lm} + \left(n_{2,lm} - 2n_{2,lm}^3 \right) v_{2,l} - 2n_{2,lm}^2 n_{1,lm} v_{1,l} \Big] \\ = \rho S_{lm} \Big[& v_{1,l} n_{1,lm} + v_{2,l} n_{2,lm} - n_{1,lm} v_{1,lm} \left(n_{1,lm}^2 + n_{2,lm}^2 \right) \\ & - n_{2,lm} v_{2,lm} \left(n_{1,lm}^2 + n_{2,lm}^2 \right) \Big] \\ = 0 \qquad & . \end{aligned} \tag{2.177}$$

Thus, the discretised form satisfies the boundary condition exactly. Because of the zero mass flux and because symmetry type boundaries are assumed to be fixed, the convection part of the momentum flux is likewise zero, i.e.

$$\rho \left(v_i \bar{v} - v_i v_g \right)_{lm} S_{lm} = 0 \, . \tag{2.178}$$

The slip and moving slip conditions are used at solid boundaries where the normal component of the flow velocity at the boundary and the normal component of the velocity of the boundary are equal. These boundary condition types specify the normal component of the velocity. For a slip boundary the normal component of the velocity is zero and for a moving slip boundary it is some constant value. However, the conditions do not specify the value of the tangential component. The normal and tangential velocity components are, therefore, treated separately. For the normal component the fixed value condition (2.170) is used, whereas the tangential component is extrapolated linearly with Eq. (2.172) or (2.173). Just as with the symmetry condition, the required operations are combined into a single operation using the Cartesian velocity components. The component forms for the slip condition become

$$\begin{aligned} v_{1,m} &= \left(1 - 2n_{1,lm}^2\right) v_{1,l} - 2n_{1,lm} n_{2,lm} v_{2,l} \\ &\quad + n_{2,lm} \varepsilon_{3ij} \frac{\partial v_{i,l}}{\partial x_k} (x_{k,m} - x_{k,l}) n_{j,lm} \\ v_{2,m} &= \left(1 - 2n_{2,lm}^2\right) v_{2,l} - 2n_{1,lm} n_{2,lm} v_{1,l} \\ &\quad - n_{1,lm} \varepsilon_{3ij} \frac{\partial v_{i,l}}{\partial x_k} (x_{k,m} - x_{k,l}) n_{j,lm} \end{aligned} \tag{2.179}$$

or

$$\begin{aligned} v_{1,m} &= \left(1 - 2n_{1,lm}^2\right) v_{1,l} - 2n_{1,lm} n_{2,lm} v_{2,l} \\ &\quad + n_{2,lm} \varepsilon_{3ij} (v_{i,l} - v_{i,lm}^{v,l}) n_{j,lm} \\ v_{2,m} &= \left(1 - 2n_{2,lm}^2\right) v_{2,l} - 2n_{1,lm} n_{2,lm} v_{1,l} \\ &\quad - n_{1,lm} \varepsilon_{3ij} (v_{i,l} - v_{i,lm}^{v,l}) n_{j,lm} \end{aligned} \tag{2.180}$$

depending on whether the Taylor extrapolation-based approximation or the special opposite node-based extrapolation is used. For moving slip condition an additional term appears, and the component forms are

$$\begin{aligned} v_{1,m} &= \left(1 - 2n_{1,lm}^2\right) v_{1,l} - 2n_{1,lm} n_{2,lm} v_{2,l} \\ &\quad + n_{2,lm} \varepsilon_{3ij} \frac{\partial v_{i,l}}{\partial x_k} (x_{k,m} - x_{k,l}) n_{j,lm} + 2n_{1,lm} v_{g,lm} \\ v_{2,m} &= \left(1 - 2n_{2,lm}^2\right) v_{2,l} - 2n_{1,lm} n_{2,lm} v_{1,l} \\ &\quad - n_{1,lm} \varepsilon_{3ij} \frac{\partial v_{i,l}}{\partial x_k} (x_{k,m} - x_{k,l}) n_{j,lm} + 2n_{2,lm} v_{g,lm} \end{aligned} \tag{2.181}$$

and

$$\begin{aligned} v_{1,m} &= \left(1 - 2n_{1,lm}^2\right) v_{1,l} - 2n_{1,lm}n_{2,lm}v_{2,l} \\ &\quad + n_{2,lm}\varepsilon_{3ij}(v_{i,l} - v_{i,lm}^{v,l})n_{j,lm} + 2n_{1,lm}v_{g,lm} \\ v_{2,m} &= \left(1 - 2n_{2,lm}^2\right) v_{2,l} - 2n_{1,lm}n_{2,lm}v_{1,l} \\ &\quad - n_{1,lm}\varepsilon_{3ij}(v_{i,l} - v_{i,lm}^{v,l})n_{j,lm} + 2n_{2,lm}v_{g,lm} \end{aligned} \tag{2.182}$$

for the Taylor extrapolation-based approximation and the opposite node-based extrapolation respectively. As the slip and moving slip type boundary conditions do not specify a condition for the pressure, it is treated identically to the inflow boundary and the ghost cell pressure is extrapolated linearly from the domain.

Similarly to the symmetry boundary condition it can be shown that the resulting mass flux for the discretised slip condition is also zero. By comparing Eqs. (2.179) and (2.180) with Eq. (2.176) it can be seen that the first lines of the equations for the slip boundary condition are the same as the equations for the symmetry condition. Thus, the first lines of the equations result in zero mass flux (see Eq. 2.177). The second lines of the equations give zero mass fluxes as well, because with Eq. (2.179)

$$\begin{aligned} \dot{m}_{lm} = \rho S_{lm}\bar{v}_{lm} = \frac{\rho S_{lm}}{2}\Bigg[& n_{1,lm}n_{2,lm}\varepsilon_{3ij}\frac{\partial v_{i,l}}{\partial x_k}(x_{k,m} - x_{k,l})n_{j,lm} \\ & - n_{2,lm}n_{1,lm}\varepsilon_{3ij}\frac{\partial v_{i,l}}{\partial x_k}(x_{k,m} - x_{k,l})n_{j,lm}\Bigg] = 0 \end{aligned} \tag{2.183}$$

and with Eq. (2.180) similarly

$$\begin{aligned} \dot{m}_{lm} = \rho S_{lm}\bar{v}_{lm} = \frac{\rho S_{lm}}{2}\Big[& n_{1,lm}n_{2,lm}\varepsilon_{3ij}(v_{i,l} - v_{i,lm}^{v,l})n_{j,lm} \\ & - n_{2,lm}n_{1,lm}\varepsilon_{3ij}(v_{i,l} - v_{i,lm}^{v,l})n_{j,lm}\Big] = 0 \,. \end{aligned} \tag{2.184}$$

Thus, the boundary condition for the mass flux is satisfied exactly in both cases. The zero mass flux and the fact that slip boundaries are fixed ensure that the momentum flux is also zero on the boundary, i.e.

$$\rho\left(v_i\bar{v} - v_i v_g\right)_{lm} S_{lm} = \rho v_{i,lm}\bar{v}_{lm}S_{lm} = 0 \,. \tag{2.185}$$

With the moving slip boundary condition the only differences to the slip condition are the last terms on the right hand sides of Eqs. (2.181) and (2.182). Thus, only these terms contribute to the mass flux. For both approaches the mass flux is

$$\dot{m}_{lm} = \rho S_{lm}\bar{v}_{lm} = \frac{\rho S_{lm}}{2}\left(2v_{g,lm}n_{1,lm}^2 + 2v_{g,lm}n_{2,lm}^2\right) = \rho S_{lm}v_{g,lm} \,, \tag{2.186}$$

which is equal to the exact value of the mass flux relative to a fixed face approximated with a midpoint rule. It should be noted that for deforming domains the continuity equation is applied for the instantaneous domain (see Eq. 2.13) and, thus, the mass flux through a moving slip boundary is non-zero. The convection part of the momentum flux (2.34) on the boundary is then easily shown to disappear, because

$$\rho \left(v_i \bar{v} - v_i v_g \right)_{lm} S_{lm} = \rho \left(v_{i,lm} v_{g,lm} - v_{i,lm} v_{g,lm} \right) S_{lm} = 0 \, . \tag{2.187}$$

The treatment of the boundary conditions on the free surface is related to the partially coupled approach discussed in Sec. 2.6. The slip condition is used in the case of the dynamic approach. With the kinematic approach the pressure on the surface is defined by the dynamic boundary condition and is thus set using Eq. (2.170). The velocity components are extrapolated into the ghost cells from the domain. Linear extrapolation is normally used for the components and the values in the ghost cells are determined with Eq. (2.172) or (2.173). Zero-order extrapolation by simply copying the components into the ghost cell, as in Eq. (2.171), might also be used. However, it has been shown that this leads to prohibitively strong damping (see Chapter 3 and e.g. Schweighofer, 2003; Mikkola, 2003).

In certain cases a more restrictive condition can be used for the extrapolation of the velocity components on the free surface. For example, in the case of non-vortical flow the zero vorticity condition

$$\frac{\partial v_1}{\partial x_2} - \frac{\partial v_2}{\partial x_1} = 0 \, . \tag{2.188}$$

can be used in the extrapolation of the velocity component, which is tangential to the face. By writing the equation in a coordinate system fixed to the boundary it becomes

$$\frac{\partial v_t}{\partial n} = \frac{\partial v_n}{\partial s} \, . \tag{2.189}$$

The extrapolation is performed in this same coordinate system using the Taylor-based extrapolation (2.172) for both velocity components. In matrix form this gives

$$\begin{pmatrix} v_t \\ v_n \end{pmatrix}_m = \begin{pmatrix} v_t \\ v_n \end{pmatrix}_l + \begin{pmatrix} \frac{\partial v_t}{\partial s} & \frac{\partial v_t}{\partial n} \\ \frac{\partial v_n}{\partial s} & \frac{\partial v_n}{\partial n} \end{pmatrix}_l \begin{pmatrix} -n_2 & n_1 \\ n_1 & n_2 \end{pmatrix}_{lm} \begin{pmatrix} x_{1,m} - x_{1,l} \\ x_{2,m} - x_{2,l} \end{pmatrix} \tag{2.190}$$

By substituting Eq. (2.189) into Eq. (2.190) and by writing it in the original coordinate

system, the component form becomes

$$\begin{aligned} v_{1,m} = v_{1,l} &+ \left[\frac{\partial v_{1,l}}{\partial x_1} + n_{1,lm} n_{2,lm} \left(\frac{\partial v_{2,l}}{\partial x_1} - \frac{\partial v_{1,l}}{\partial x_2} \right) \right] (x_{1,m} - x_{1,l}) \\ &+ \left(\frac{\partial v_{1,l}}{\partial x_2} n_{1,lm}^2 + \frac{\partial v_{2,l}}{\partial x_1} n_{2,lm}^2 \right) (x_{2,m} - x_{2,l}) \\ v_{2,m} = v_{2,l} &+ \left[\frac{\partial v_{2,l}}{\partial x_2} - n_{1,lm} n_{2,lm} \left(\frac{\partial v_{2,l}}{\partial x_1} - \frac{\partial v_{1,l}}{\partial x_2} \right) \right] (x_{2,m} - x_{2,l}) \\ &+ \left(\frac{\partial v_{1,l}}{\partial x_2} n_{1,lm}^2 + \frac{\partial v_{2,l}}{\partial x_1} n_{2,lm}^2 \right) (x_{1,m} - x_{1,l}) \, . \end{aligned} \tag{2.191}$$

The problem with extrapolation at the boundaries is that the values extrapolated into the ghost cells depend on gradients, which, on the other hand, depend on the values in the ghost cells. By using the form (2.191), the influence of the gradient component normal to the face is reduced in the extrapolation of the tangential component.

2.7.2 The implicit treatment

The discussion in the previous section is relevant from the point of view of those parts of the solution algorithm where the actual values of flow variables outside the domain are required, namely, the calculation of the explicit momentum fluxes and the mass fluxes. However, the discretisation of the boundary conditions for the implicit stage of the momentum calculation and for the pressure correction equation have to be treated separately.

The implicit momentum equations are based on the linearisation of the momentum fluxes. Thus, the treatment of the boundary conditions in the implicit stage is directly linked to the discretisation of the conditions for the explicit fluxes. If the boundary condition specifies the velocity components on the boundary directly, in the implicit stage the change in the velocity on the face is set to zero. By linearising Eq. (2.170) with respect to the variables of the equation and by realising that $\Delta\phi_{lm} = 0$, the boundary condition is satisfied by setting

$$\Delta v_{i,m} = -\Delta v_{i,l} \, . \tag{2.192}$$

If the boundary condition specifies velocity components through some derived quantity or quantities, such as in the case of the symmetry or the slip condition, the velocity components may be coupled. As the velocity components are treated separately, for the sake of simplicity the changes in the velocity components in a ghost cell are set to zero in the

implicit stage. This does not influence the final solution, as in steady state and at the end of each physical time step iteration the changes throughout the domain and at the boundary of the domain should vanish.

If the boundary condition does not specify the velocity at the boundary, either Eq. (2.172) or (2.173) is used to extrapolate the value in the ghost cell. The linearisation of either of these would involve a large stencil of neighbouring cell centres. Therefore, the linearisation is based on zero-order extrapolation, i.e. on the assumption that the value of the change in the ghost cell is equal to the value in the physical cell centre. The implicit boundary condition is then

$$\Delta v_{i,m} = \Delta v_{i,l} \; . \tag{2.193}$$

The treatment of the boundary conditions with the pressure correction equation is very similar to the approach with the implicit momentum equations. If the pressure on the face is specified by the boundary condition, the pressure correction on the face is zero and the value in the ghost cell is set as in Eq. (2.192). Similarly, if the boundary condition does not specify the pressure at the boundary and, thus, the pressure is extrapolated into the ghost cell from the computational domain in the explicit stage, a zero-order extrapolation is used in the pressure correction equation. Thus, just as in Eq. (2.193), the pressure correction in the ghost cell is assumed to be equal to that in the physical cell centre. The latter condition is also used in the case of the Neumann boundary condition for the pressure. This is equal to the assumption of zero mass flux correction on the face, as can be seen from Eq. (2.149). This is applied, for example, in the case of the mirror boundary condition. In the case of free surface boundaries the pressure correction is specified and the pressure correction in the ghost cell is set according to Eq. (2.164), as described in Sec. 2.6.

By looking at the implicit boundary conditions it can be seen that either the change in the ghost cell is zero or it is given in terms of the change in the corresponding physical cell. In the solution of the momentum or pressure correction equations the implicit boundary conditions are taken into account by transferring the off-diagonal contribution from the ghost cell into the diagonal element of the corresponding physical cell. Thus, the solution of the linear systems is performed over the physical cells, and the ghost cell values are updated only after a solution for the physical cells is obtained.

2.8 Grid updating

As the boundaries of the computational domain translate and deform, the grid has to be updated to match the new boundaries and to maintain adequate grid quality. Several methods exist for the updating of the grid, ranging from simple algebraic approaches, through discrete and continuous structural analogies, to grid regeneration. In this work the updating of the interior node locations is based on a linear spring analogy (Batina, 1991) with optional torsional springs (Farhat et al., 1998), a simple algebraic smoothing process (Mavriplis and Jameson, 1987; Anderson, 1992), or a combination of these. The interior points are updated according to the node displacements at the boundaries. The nodes at the boundaries are relocated and redistributed on the translating and deforming piecewise linear boundary definitions with the spacing of the nodes based on the original node distribution (see Sec. 2.3.2).

An overview of the steps for the updating of the grid at each iteration is given below. The practical implementation of the boundary and interior node updates is discussed separately in more detail in the following sections.

1. Update the boundary geometry.
    a. At the start of every pseudo-iteration, the free surface geometry definition is updated by adding the change in the wave height components to the previous values.
    b. At the start of every physical time step, time-dependent boundaries are updated by updating the respective geometry definitions.
2. Search for the new intersections of each boundary on the basis of the updated geometry definitions.
3. Redistribute the grid points on the updated geometries on the basis of the new intersection points and the original point distribution.
4. In the case of rigid movement (rotation or translation), move all boundary grid points in the opposite direction (only at the start of a physical time step).
5. Update the grid by solving the equilibrium equations for the linear/torsional spring system.
6. Perform Laplacian smoothing for the grid.

7. In the case of rigid movement, perform the movement for the whole grid, i.e. for both the boundary and the interior nodes (only at the start of a physical time step).

Because of steps 4 and 7, rigid transformations are actually performed in a coordinate system fixed to the body. This ensures that the grid points close to the moving body are not translated or translated only slightly by the spring system and helps to preserve the original quality of the grid close to the body, even in cases of large body movements. However, the approach as such only applies to cases in which there is just a single body. In the case of several bodies, some kind of weighting could be applied (see e.g. Leroyer and Visonneau, 2003). It should also be noted that rigid transformations are performed only at the start of each physical time step. During the pseudo-iterations, from the point of view of grid updating, only free surface boundaries deform and other moving boundaries are fixed.

2.8.1 The updating of the boundary nodes

The core of the boundary node update corresponds to steps 2 and 3 in the list above. The process starts with the definition of the new locations of the intersections of different boundary segments. For a particular pair of boundary sections the intersecting linear segments on both boundary definitions are first searched for. This is done by traversing through the segments on both boundaries at the same time and by monitoring the signs of the cross-products (see Fig. 2.5 for the node definitions)

$$\begin{aligned} r_{11} &= \varepsilon_{3ij}(x_{i,l+1}^{B_{j+1}} - x_{i,k}^{B_j})(x_{j,l}^{B_{j+1}} - x_{i,k}^{B_j}) \\ r_{12} &= \varepsilon_{3ij}(x_{i,l+1}^{B_{j+1}} - x_{i,k+1}^{B_j})(x_{j,l}^{B_{j+1}} - x_{i,k+1}^{B_j}) \\ r_{21} &= \varepsilon_{3ij}(x_{i,k}^{B_j} - x_{i,l}^{B_{j+1}})(x_{j,k+1}^{B_j} - x_{i,l}^{B_{j+1}}) \\ r_{22} &= \varepsilon_{3ij}(x_{i,k}^{B_j} - x_{i,l+1}^{B_{j+1}})(x_{j,k+1}^{B_j} - x_{i,l+1}^{B_{j+1}}) \ . \end{aligned} \tag{2.194}$$

The search in sections B_j and B_{j+1} can be terminated when the products $r_{11}r_{12},\ r_{21}r_{22} \leq 0$.

The boundary definition curves are constructed in such a way that they generally intersect if both boundaries are other than free surface boundaries. In this case the intersecting segments are easily found by the algorithm. The case is illustrated in Fig. 2.19. If one of the boundaries is a free surface boundary, the definition curves do not generally intersect, as the definition is based on the wave height coordinates and as these are defined at the

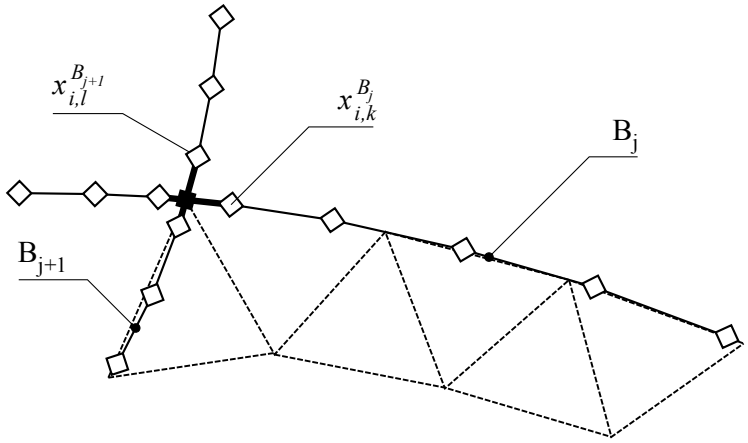


Figure 2.19: If the boundary definition curves intersect, the intersection point (solid square) is found by searching for the intersecting segments (bold lines) at both boundaries and by determining the intersection of these two lines.

centres of the faces. Thus, two intersecting segments are not found. In this case the search algorithm terminates at the end of the boundary definition or at the segment intersecting with the linear extension of the other boundary definition. This is illustrated in Fig. 2.20 The intersection of the two linear sections is always determined by solving the equation

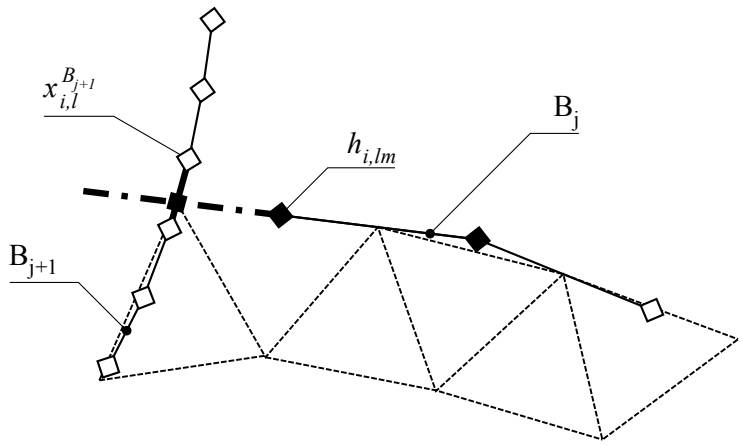


Figure 2.20: If the boundary definition curves do not intersect, the intersection point is found by extending (dash-dot line) one or both curves with a line through the last two points (solid diamonds) of the definition curve.

$$x_{i,k}^{B_j} + a(x_{i,k+1}^{B_j} - x_{i,k}^{B_j}) = x_{i,l}^{B_{j+1}} + b(x_{i,l+1}^{B_{j+1}} - x_{i,l}^{B_{j+1}}) \,, \tag{2.195}$$

where a and b are the coefficients to be solved.

After the new intersection points are found, the grid nodes are distributed on the updated boundary definitions. A slightly different implementation of step 3 is used for free surface boundaries and the rest of the boundaries. In both cases the new length of the boundary section is first evaluated on the basis of the updated intersection points. The original spacing distribution is then updated by multiplying it by the ratio of the new and the original section lengths. The rest of the treatment differs between free surface boundaries and the rest of the boundaries. On the free surface the geometry is first defined with cubic splines through the current wave coordinates. After this the wave coordinates are redistributed on the cubic spline definition on the basis of the original distribution. For the rest of the boundaries the treatment is more straightforward. The geometry definition of a particular boundary is traversed through from the intersection point at the start to the intersection point at the end of the patch, inserting grid points on the piecewise linear boundary definition based on the updated point distribution.

2.8.2 The updating of the interior nodes

As mentioned at the beginning of the section, the interior nodes of the grid are updated on the basis of a linear spring analogy with optional torsional springs, algebraic smoothing, or a combination of these. The steps of the last option were presented in the overview at the beginning of this section. In each alternative the general approach is the same, i.e. a solution for a system of equations with specified boundary node displacements is searched for. For this an iterative approach based on either the Jacobi iteration or the Dulikravitch and Huang (1987) relaxed Jacobi algorithm favoured and recommended by Farhat et al. (1998) is used. It is not necessary to find an exact solution to the equation system, but a small number of iterations is generally enough to maintain an acceptable grid quality.

Before going into the details of the different grid updating methods, some general aspects are discussed. The choice of the best alternative for interior node updating depends on the case at hand. The computational cost of grid updating increases with the complexity of the approach. In this regard, the most effective alternative is algebraic smoothing. The linear spring analogy is slightly heavier, and the inclusion of torsional springs adds further to the computational complexity of the algorithm. The computational cost of the algorithms used seems to be directly proportional to the robustness of the method. For example, although computationally efficient, smoothing contracts the grid close to convex body boundaries. In the case of a small curvature of the boundary, this may

become problematic. Furthermore, the smoothing process may deform the grid, even if the boundaries are not deformed. The linear spring analogy is more robust in this regard. On the other hand, the linear spring analogy cannot prevent a node crossing an edge. In this case added robustness is obtained with the use of torsional springs. Despite the shortcomings of each approach, with a proper choice very satisfactory results have been obtained for every alternative. The details of the different alternatives are described below.

Spring analogy

In the spring analogy the grid is assumed to consist of a network of springs. The edges of the cells are assumed to be linear springs and, in the case of the torsional spring analogy, additional torsional springs are assumed to be located at each corner of a cell. The grid updating is performed by searching for the equilibrium of the spring system. A schematic of the approach is shown in Fig. 2.21.

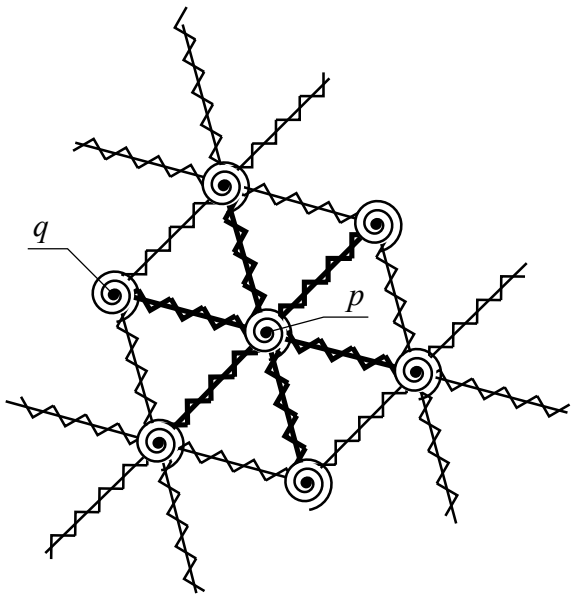


Figure 2.21: In the spring analogy the edges of the grid are assumed to be linear springs and optional torsional springs are located at the grid vertices between neighbouring edges (denoted in the figure by a single torsional spring). The new location of a grid point $p \in \mathrm{N}$ is determined by searching for the equilibrium of the system formed by the linear springs connecting the node and the neighbouring nodes $q \in \mathrm{N}_{\mathrm{p}}^{\mathrm{nb}} \subset \mathrm{N}$ and the torsional springs attached between the linear springs.

Blom (2000) distinguishes two spring analogies associated with linear springs, which he refers to as the vertex method and the segment method. In the former the equilibrium length is zero, whereas in the latter, first proposed by Batina (1991), the equilibrium lengths of the springs are their initial lengths. In this work the linear spring analogy is based on the general concept proposed by Batina. This is further extended with optional torsional springs to avoid grid crossover, as suggested by Farhat et al. (1998). The spring analogy is based on the search for node displacements that satisfy the force equilibrium of the spring system at the same time for each interior node.

The linear springs used by Batina lead to the equation

$$X_{i,p} = \sum_q k_{pq}(\delta x^v_{i,q} - \delta x^v_{i,p}) \;, \quad q \in \mathrm{N}_\mathrm{p}^\mathrm{nb} \;, \tag{2.196}$$

where $X_{i,p}$ are the Cartesian components of the force exerted on the node p, $k_{pq} = k_{qp}$ are the stiffnesses of the linear springs connecting the node p and the neighbouring nodes q, and δx_i are the components of the node displacements. In equilibrium state the force X at each node is zero. With Jacobi iteration this leads to the iterative equation

$$\delta x^{v,k+1}_{i,p} = \frac{\sum_q k_{pq} \delta x^{v,k}_{i,q}}{\sum_q k_{pq}} \;, \quad q \in \mathrm{N}_\mathrm{p}^\mathrm{nb} \;, \tag{2.197}$$

where, in this case, k denotes the Jacobi iteration step. In order to avoid node collision, Batina defines the spring stiffness as the inverse of the spring length, i.e.

$$k_{pq} = \frac{1}{\sqrt{(x^v_{1,q} - x^v_{1,p})^2 + (x^v_{2,q} - x^v_{2,p})^2}} \;. \tag{2.198}$$

In this work a modified spring stiffness equal to the square of the stiffness which Batina proposed, is used; i.e.

$$k_{pq} = \frac{1}{(x^v_{1,q} - x^v_{1,p})^2 + (x^v_{2,q} - x^v_{2,p})^2} \;. \tag{2.199}$$

This amplifies the increase in the stiffness with decreasing spring length. A similar choice was made by Blom, with the exception that he multiplied the stiffness by 5 for the first layer of nodes adjacent to the boundary.

The linear spring methodology avoids the collision of nodes by increasing the spring stiffness, as the spring length decreases, but does not prevent a node from crossing an edge. The crossover will lead to inverted cells, making the grid invalid. For this reason

Farhat et al. (1998) proposes the use of additional torsional springs. These are located at the vertices and are connected between two neighbouring edges meeting at the vertex (see Fig. 2.21). The number of torsional springs at a vertex is thus equal to the number of control volumes sharing the vertex. Farhat et al. devised the torsional spring analogy for grids composed of triangles as an extension for codes based on the linear spring analogy and it is, therefore, easy to implement in existing codes.

The general idea is similar to the one in the linear spring analogy, i.e. the spring stiffness changes as a function of the angle between the neighbouring edges. The stiffness is given by the equation

$$C_p^{pqr} = \frac{1}{\sin^2 \theta_p^{pqr}} \,, \tag{2.200}$$

where the triplet pqr refers to the three nodes forming a triangle, p refers to the spring between the edges pq and pr, and θ_p^{pqr} refers to the angle between these edges. The equilibrium equations are based on the superposition of the linear and torsional springs, in which the influence of torsional springs is represented by an equivalent force. In the method used in this work the equations presented by Farhat et al. are implemented in their original form. The interested reader is referred to the original paper, in which the full derivation and the final equations are presented.

Grid smoothing

The algebraic smoothing process first proposed by Mavriplis and Jameson (1987) is an iterative, Laplacian-type procedure, which is based on the difference between the current location of a node and the average of the locations of the neighbouring nodes. In this the interior node locations are updated from

$$x_{i,p}^{v,k+1} = x_{i,p}^{v,k} + \frac{\omega}{n(\mathrm{N_p^{nb}})} \sum_q (x_{i,q}^{v,k} - x_{i,p}^{v,k}) \,, \quad q \in \mathrm{N_p^{nb}} \,, \tag{2.201}$$

where ω is a relaxation factor and $n(\mathrm{N_p^{nb}})$ is the number of the nodes q connected to the node p. The equation is iterated until the node displacements are sufficiently converged. In this work the iteration is stopped when the L_2-norm of the node displacements is below 10^{-6} or when a specified number of iterations has been reached. Usually, the former condition is preferred and the number of iterations is set in such a way that the first condition dominates.

It is interesting to note that with a particular choice of parameters the smoothing process

is equal to the vertex spring method discussed by (Blom, 2000). The grid update equation for the vertex method is

$$x_{i,p}^{v,k+1} = \frac{\sum_q k_{pq} x_{i,q}^{v,k}}{\sum_q k_{pq}} , \quad q \in \mathrm{N_p^{nb}} , \tag{2.202}$$

In the vertex method the stiffness is taken as a constant. As in this case the numerical value does not have any influence, it can be chosen as $k_{pq} \equiv 1$. With this choice Eq. (2.202) can be written as

$$x_{i,p}^{v,k+1} = \frac{1}{n(\mathrm{N_p^{nb}})} \sum_q x_{i,q}^{v,k} , \quad q \in \mathrm{N_p^{nb}} . \tag{2.203}$$

By adding and subtracting $x_{i,p}^{v,k}$ this can be written in the form

$$x_{i,p}^{v,k+1} = x_{i,p}^{v,k} + \frac{1}{n(\mathrm{N_p^{nb}})} \sum_q (x_{i,q}^{v,k} - x_{i,p}^{v,k}) , \quad q \in \mathrm{N_p^{nb}} , \tag{2.204}$$

By comparing Eqs. (2.201) and (2.204) it can be seen that the vertex method is equal to grid smoothing with a relaxation factor $\omega = 1$. However, in grid smoothing smaller relaxation factors are typically used. For example, Anderson (1992) used $\omega = 0.2$, whereas in this work $\omega = 0.5$ is used.

3 Discussion on the coupling of the bulk flow and the free surface solutions

Several alternative free surface approaches have been implemented in the solution method. These are related, for example, to the coupling of the bulk flow and free surface solutions (see Sec. 2.6). Before the verification of the method that was developed was commenced, some qualitative comparisons were performed in order to assess the performance of the different coupling approaches. These comparisons reveal significant differences mainly in the robustness and efficiency of the alternatives and these are discussed in the following section. On the basis of these comparisons a single coupling approach has been selected for the code verification study in Chapter 6.

The comparisons are not meant to be exhaustive, but the purpose is to find and demonstrate in practical terms the most salient differences between the approaches. For example, the simulations were performed with single grids and, thus, the influence of the grid on the solutions and the related error is not considered here. Furthermore, the approaches are compared mainly in terms of their ability and efficiency to run the solution changes to negligible level, i.e. to find the converged solution. In order to do this, generally each case has been iterated to the limit of machine accuracy (single precision), if possible. Because of this, the error related to incomplete iteration is not considered either. These solution related errors are discussed exclusively in Part II.

3.1 The test cases

The approaches to the coupling of the bulk flow and free surface solutions described in Sec. 2.6 were studied with two steady test cases (Mikkola, 2003). Additionally, the kinematic approach was studied with an unsteady case in order to investigate the influence of the coupling with unsteady flows. The steady cases are a flow over a bump on the bottom of an infinitely wide channel Bet et al. (1996) and a flow over a submerged hydrofoil with an angle of attack Duncan (1983). For the unsteady study, the manufactured solution test case based on a standing wave in a rectangular container was used. The particulars of the steady cases are discussed in detail below. The details of the unsteady case are covered only briefly below, as the full description of the case can be found in Chapter 6.

The steady cases were simulated with three different coupled free surface solution approaches and for comparison also with fully decoupled approaches. The results are presented for:

a. the kinematic free surface approach (Sec. 2.6.1) with zero gradient condition (2.171) for the velocity components;

b. the kinematic free surface approach (Sec. 2.6.1) with linear extrapolation (2.173) for the velocity components;

c. the dynamic free surface approach (Sec. 2.6.2);

d. the fully decoupled kinematic approach which is identical to case b, but Eq. (2.163) is replaced with condition $p'_{lm} = 0$.

A fully decoupled dynamic approach which is identical to case c, but without the additional contribution (2.169) to the pressure correction equation, was tested as well. However, due to significant convergence problems no results are available for this case. The convergence behaviours of the two coupled kinematic cases (Cases a and b) are not presented separately, as only minor differences were observed between the cases. Thus, case b has been chosen for the convergence comparisons.

It should be noted that because of Eq. (2.165) wave heights are under-relaxed also in case of the fully decoupled approach. The amount of under-relaxation depends on the value of the under-relaxation factor α_p for the pressure corrections. The choice of under-relaxing wave heights also in the decoupled approach was made to have comparable wave height updates in both the coupled and the decoupled kinematic approaches. However, in the coupled kinematic approach the wave height updates are passed to the pressure correction equation without under-relaxation and the relaxation is applied only after the pressure correction step. Because of this, some caution should be exercised, when comparing the performance of different approaches.

The steady cases were simulated using the SIMPLE method for the pressure correction equation. In both steady cases, the opposite node-based approach (2.173) was used for extrapolation-type boundary conditions. However, different discretisations were used for the momentum flux. For the flow over a bump the Frink extrapolation for triangles (2.46) was used, whereas in the hydrofoil case the convected velocity components were evaluated with the Taylor extrapolation-based approach (2.45).

In the unsteady study the pressure correction equation based on the SIMPLEC method was used and, thus, no under-relaxation was required for the pressures and the free surface deformations. The choice of the discretisations and the boundary conditions in this case is discussed in Sec. 6.2.

3.1.1 Flow over a bump

This case was chosen for the overall comparison of different approaches during the development of the free surface solution method because of its simplicity. Testing was performed with both the unstructured solver described in this thesis and with a structured counterpart. The findings from the test performed with these two different solvers were, however, very similar and thus only the results for the unstructured versions are presented.

The geometry of the bump on the floor of the channel is given by

$$x_2 = -1 + 0.1e^{-(x_1-10)^2} \tag{3.1}$$

with a height of 0.1 and the centre at $x_1 = 10$. In the x_1 direction the domain extends from -20 to 50, with damping zones from -20 to 0 and from 30 to 50 on the free surface. In the damping zones first-order upwind discretisation (2.65) was used for the discretisation of the free surface slope. The undisturbed water level was at $x_2 = 0$. The grid for the case was created with the EasyMesh grid generator (Niceno) and has 1533 nodes, 4123 sides, and 2591 triangles. The number of points on the free surface is 279. The cell size in the grid increases towards the inflow and outflow boundaries in order to minimise wave reflection. Grid updating was performed using the linear spring analogy. The results presented below are for the sub-critical case with a depth-based Froude number of 0.567. Qualitative observations of super-critical cases are also given.

3.1.2 Flow over a submerged hydrofoil

The accuracy and the convergence behaviour of the free surface approaches have been checked further with a flow over a submerged NACA-0012 hydrofoil with an angle of attack of 5 degrees. The Froude number, based on the chord length, is 0.567. Experimental results for this case were presented by Duncan (1983). It has also been extensively used for numerical testing (see e.g. Hino et al., 1993; Lungu and Mori, 1993).

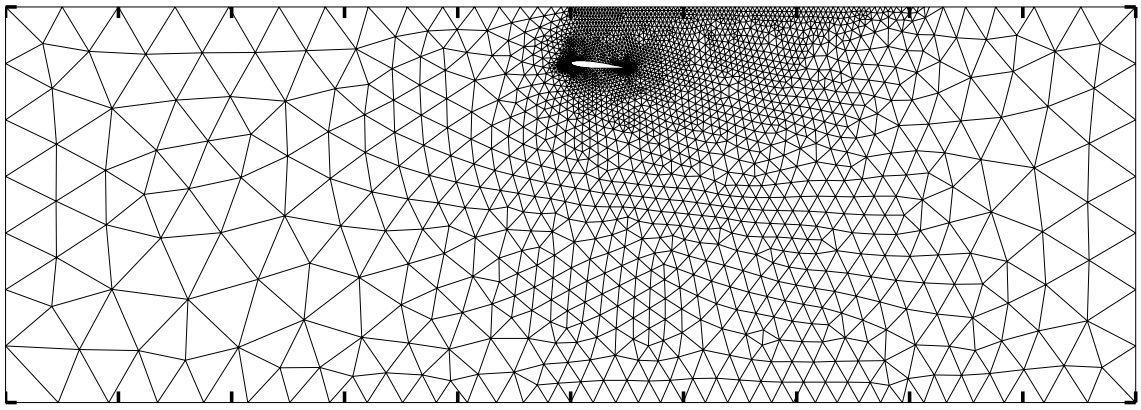


Figure 3.1: Computational domain and unstructured grid for flow over a submerged hydrofoil.

The chord length of the hydrofoil is 1 and the leading edge is at (0,-0.99). The computational domain $[-10, 10] \times [-7, 0]$ and the unstructured grid generated for the case with EasyMesh are presented in Fig. 3.1. The grid consists of 2996 points, 8699 sides, and 5703 triangles. There are damping zones on the free surface from the inflow boundary to -7 and from 6.25 to the outflow boundary. The resolution of the grid is increased around the leading and trailing edges of the foil, as well as close to the free surface between $x_1 = 0$ and $x_1 = 6.25$. The number of points on the hydrofoil and on the free surface is 128 and 115 respectively. Just as with the first test case, the grid updating was based on the linear spring analogy.

3.1.3 Flow for a standing wave

After the studies with the steady cases, the investigation on the influence of the solution coupling was extended to cover an unsteady case as well. On the basis of the steady results and because the simulation of unsteady flows was not implemented into the fully decoupled counterpart of the dynamic approach, the study on the influence of the coupling is restricted to the kinematic approach. For simplicity, the case is the same as the one used in the unsteady MMS study (see Chapter 6). For the simulations the coarsest spatial and three coarsest temporal discretisations were chosen (see Tab. 6.1 on page 166).

3.2 The analysis of the convergence histories for the steady cases

In order to compare the behaviour of the different approaches in the steady cases the convergence histories of the L_2-norm of the change of velocity components and the pressure were analysed. The analysis is based on an approximate reconstruction of the most prominent features of the convergence histories. A typical convergence history is presented in Fig. 3.2.

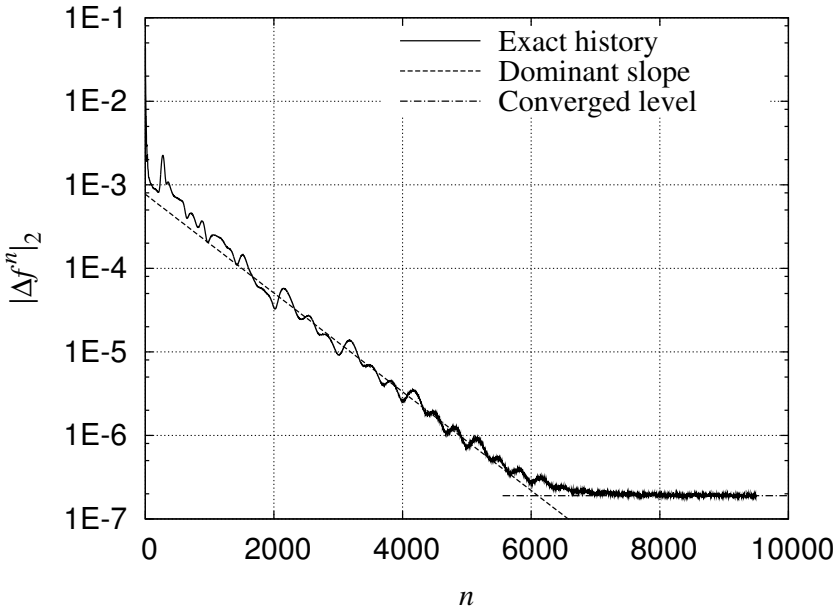


Figure 3.2: A typical convergence history. The dominant part of the history is approximated with a linear curve on a semi-logarithmic scale.

It can be seen from the figure that the dominant part of the convergence history of a variable can be approximated on a semi-logarithmic scale by a linear curve. Thus, for a general variable f the convergence of the L_2-norm of the change in the dominant part is approximated as

$$|\Delta f^n|_2 \approx |\Delta \tilde{f}^0|_2 (G_f)^n \,, \tag{3.2}$$

where G_f is the approximate ratio of values at successive pseudo iterations and $\Delta \tilde{f}^0$ is a parameter corresponding to the value of the approximation at the iteration $n = 0$. The parameters of the approximation were determined by a least-squares fit for

$$\ln |\Delta f^n|_2 \approx n \ln(G_f) + \ln |\Delta \tilde{f}^0|_2 \,. \tag{3.3}$$

It should be emphasised that Eq. (3.2) is not an accurate construction of the entire convergence history, but attempts to characterise the convergence behaviour in a general sense. Particularly, the characteristics of the convergence history for the initial and the dominant stages differ significantly. Thus, $\Delta\tilde{f}^0$ does not approximate the convergence at the beginning of the iteration, but is merely an auxiliary parameter of the approximate reconstruction.

The L_2-norms for the converged solutions are approximated by taking the average of a convergence history over the converged part, i.e.

$$|\Delta\tilde{f}^\infty|_2 \approx \lim_{n\to\infty} |\Delta f^n|_2 \,, \tag{3.4}$$

where $|\Delta\tilde{f}^\infty|_2$ is the average. In most cases this part corresponds to a solution which has converged to machine accuracy, but in some cases the convergence settles at a level higher than the machine accuracy.

Because it is difficult to see the practical implications of a change in the convergence rate directly using just the ratio G_f, e.g. in terms of the number of iterations required to reach a specific convergence level, two additional measures for the convergence rate are used in the analysis of the simulation results. The first one approximates the number of iterations required to drop the L_2-norm of a variable change by one order of magnitude. On the basis of Eq. (3.2), this can be approximated as

$$\tilde{N}_f^{1/10} = \frac{\ln 0.1}{\ln G_f} \,. \tag{3.5}$$

The second measure is based on the number of iterations required to reach converged solution based on the approximate construction (3.2) which is defined as

$$\tilde{N}_f^{\infty} = \frac{\ln |\Delta\tilde{f}^\infty|_2 - \ln |\Delta\tilde{f}^0|_2}{\ln G_f} \,. \tag{3.6}$$

Because of the smooth transition between the dominant part and the converged part in the convergence histories (see Fig. 3.2), this is not an accurate measure of the actual number of iterations required to reach a converged solution. However, it has been observed that the actual number of iterations is higher by a constant factor (between 1.2 and 1.3). Thus, the actual number of required iterations is directly proportional to the approximate number of iterations. As in the comparisons the relative differences between approaches rather than the absolute values are of interest, the use of Eq. (3.6) as the basis of the comparison is justified.

3.3 The results for the flow over a bump

The first steady test case was used to study the stability characteristics of the different coupling approach. The case was simulated with a set of pressure under-relaxation factors and free surface timesteps using the different alternatives. For the pressure under-relaxation factor two alternative values (0.2 and 0.3) have been used. The timestep parameter for the bulk flow $C_{\Delta\tau}$ was fixed at 5.0, whereas the parameter for the free surface pseudo time integration $C_{\Delta\tau^f}$ was varied. The range of time step parameters depends on the case. Initially, each case was simulated with $C_{\Delta\tau^f} = 0.5$, 1.0, 1.5 and 2.0. On the basis of these results additional time step values (with increments of 0.1) were also tested, in order to find the largest time step still resulting in convergence to machine accuracy. In the following the analysed results are presented for the converging cases of the initial set $\{0.5, 1.0, 1.5, 2.0\}$ as well as for both the largest time step with convergence to machine accuracy and the successive time step (see Sec. 3.2 for the analysis procedure). The complete results of the analysis are presented in Appendix A.

The ratios of successive values G_f and the convergence level $|\Delta\tilde{f}^{\infty}|_2$ for the velocity components and the pressure are shown in Tab. 3.1. By comparing the results for the coupled and the decoupled kinematic approach (Cases b and d) with the smaller pressure under-relaxation factor it can be observed that in this case the coupling does not have any influence on the largest allowable time step. However, with the coupled approach (Case b) this time step results in significantly faster convergence. In this case the ratios of succesive values G_f correspond to a drop of one order of magnitude in 1890 and 2990 iterations for the coupled and the decoupled approaches respectively. With the larger pressure under-relaxation factor the influence of the coupling in the kinematic approach is obvious. In this case the decoupled approach (Case d) is more unstable, and the time step had to reduced significantly. On the other hand, with the coupled approach the time step had to be reduced only slightly. Furthermore, in case of the coupled approach even if the time step was reduced, faster convergence was obtained with the reduced time step (one order of magnitude drop in 1680 iterations). With the decoupled approach the reduction of the time step was so significant that it also reduced the convergence speed (one order of magnitude drop in 4640 iterations).

In the case of the dynamic approach the decoupled approach was extremely unstable and, even with radical reduction of the pressure under-relaxation factor, the computations diverged. Here, the coupling has a significant influence on the stability of the approach. The testing showed that the coupled approach (Case c) is stable, i.e. does not diverge,

Table 3.1: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a bump. Top: the coupled kinematic approach (Case b); middle: the coupled dynamic approach (Case c); bottom: the decoupled kinematic approach (Case d). The lines with italic entries refer to cases, which failed to converge to the machine accuracy.

	α_p	$C_{\Delta\tau^f}$	G_u	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^\infty\rvert_2$
Case b	0.2	0.5	.99932	1.8E-7	.99933	6.5E-8	.99933	3.6E-4
	0.2	1.0	.99896	1.9E-7	.99896	6.7E-8	.99895	3.7E-4
	0.2	1.3	.99878	1.9E-7	.99879	6.9E-8	.99878	3.7E-4
	0.2	*1.4*	*.99869*	*1.9E-7*	*.99870*	*7.1E-8*	*.99870*	*3.8E-4*
	0.3	0.5	.99913	1.9E-7	.99913	7.2E-8	.99913	4.3E-4
	0.3	1.0	.99872	1.9E-7	.99870	7.5E-8	.99869	4.4E-4
	0.3	1.1	.99863	1.9E-7	.99863	7.7E-8	.99862	4.4E-4
	0.3	*1.2*	*.99857*	*1.9E-7*	*.99856*	*8.0E-8*	*.99856*	*4.5E-4*
Case c	0.2	0.5	.99932	3.3E-7	.99939	7.4E-8	.99936	3.7E-4
	0.2	0.7	.99914	3.0E-7	.99919	7.1E-8	.99917	3.7E-4
	0.2	*0.8*	*.99907*	*2.9E-7*	*.99912*	*7.2E-8*	*.99909*	*3.8E-4*
	0.3	0.4	.99921	3.6E-7	.99927	8.4E-8	.99924	4.3E-4
	0.3	*0.5*	*.99906*	*3.3E-7*	*.99913*	*8.1E-8*	*.99910*	*4.3E-4*
Case d	0.2	0.5	.99967	1.9E-7	.99967	6.7E-8	.99967	3.9E-4
	0.2	1.0	.99950	1.9E-7	.99949	7.2E-8	.99949	4.2E-4
	0.2	1.3	.99923	1.9E-7	.99923	7.7E-8	.99923	4.9E-4
	0.2	*1.4*	*.99912*	*2.1E-7*	*.99913*	*1.1E-7*	*.99913*	*7.6E-4*
	0.3	0.5	.99952	2.0E-7	.99952	8.3E-8	.99952	5.4E-4
	0.3	0.6	.99951	2.0E-7	.99950	8.4E-8	.99950	5.6E-4
	0.3	*0.7*	*.99947*	*2.4E-7*	*.99947*	*1.5E-7*	*.99947*	*1.1E-3*

with time steps that are comparable to the kinematic approach. However, compared to the coupled kinematic approach (Case b) considerably smaller time steps had to be used in order to get convergence to machine accuracy. The pressure under-relaxation factor had a similar effect as in the other cases, i.e. with increasing under-relaxation factor the time step had to be reduced. The notable difference compared to the kinematic approaches is the variation of the ratios of successive values G_f for different flow variables. For the fastest convergence ($\alpha_p = 0.2$, $C_{\Delta\tau f} = 0.7$) the number of iterations for one order of magnitude drop range from 2680 to 2840 depending on the flow variable. With both kinematic approaches practically identical values are obtained regardless of the flow variable. The variation of the values is caused by the slightly more irregular character of the convergence histories with the coupled dynamic approach.

With super-critical cases considerable differences emerged between the approaches. As the inflow velocity was increased, the dynamic approach became highly unstable, with large free surface deformations leading to the divergence of the solution, unless the final pressure corrections were heavily under-relaxed. The kinematic approaches, on the other hand, required very few changes to the input parameters.

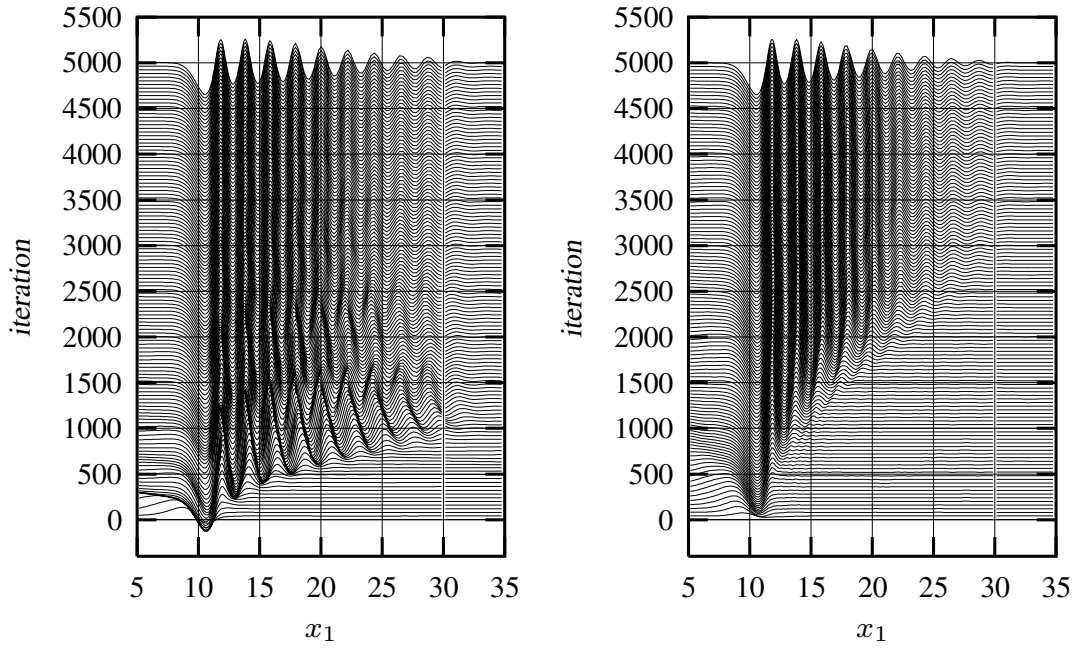


Figure 3.3: The convergence of the wave elevation in Cases b (kinematic, left) and c (dynamic, right) for the flow over a bump ($\alpha_p = 0.2$, $C_{\Delta\tau f} = 1.0$).

Fig. 3.3 compares the development of the wave field between the kinematic and the dy-

namic approach in the sub-critical case. It can be seen that the development downstream is faster for the kinematic approach. However, some oscillations in this case can be observed, particularly for the first two to three thousand iterations. These may be due to reflected waves from the initial transient. The build-up of the waves is also somewhat different. In the kinematic approach the developing wave field becomes compressed upstream with the iterations. In the dynamic approach, on the other hand, the wave field expands downstream with an increasing wave length as the solution progresses.

Some differences between the approaches can also be seen in the converged wave profiles, which are compared in Fig. 3.4 with the results of Bet et al. (1996) and from a previous test with the FINFLO flow solver (Mikkola, 1999). The damping with the dynamic approach

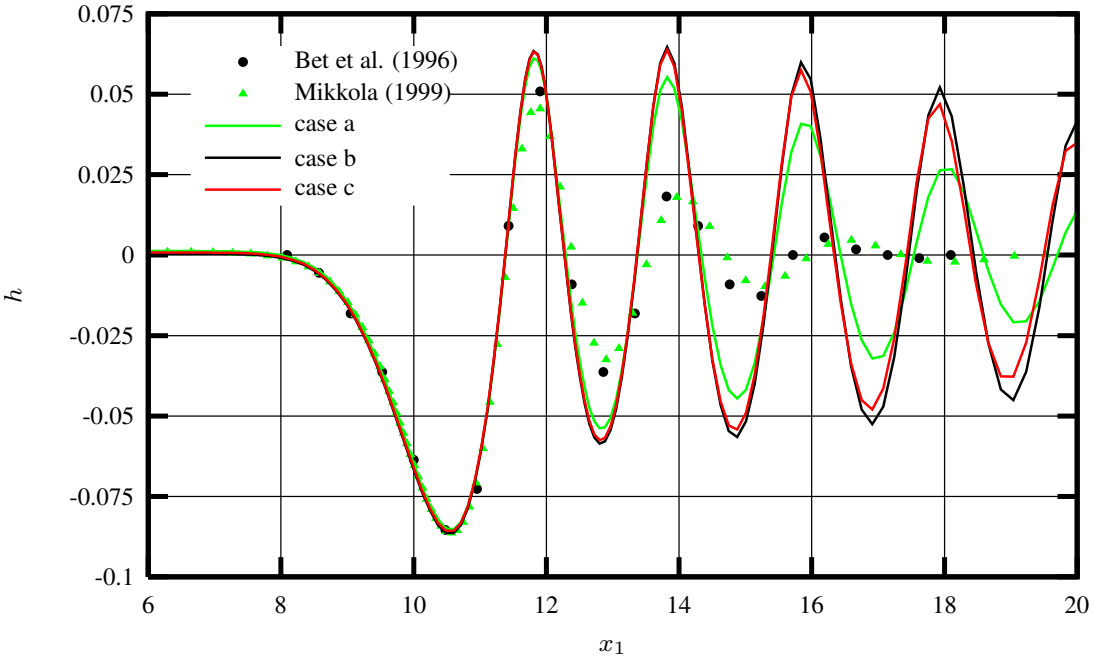


Figure 3.4: Wave profiles for flow over a bump compared with the results by Bet et al. (1996) and with the FINFLO flow solver (Mikkola, 1999). Case a: the kinematic approach with zero gradient extrapolation; case b: the kinematic approach with linear extrapolation; case c: the dynamic approach.

(Case c) is slightly stronger than with the kinematic approach and linear extrapolation for the velocity components (Case b). For the kinematic approach with a zero gradient condition (Case a), the damping is considerably stronger than with Cases b and c. This was also observed by Schweighofer (2003). The location and depth of the first through, as well as the location of the first peak for all cases, agree well with the previous results for

the structured FINFLO solver as well as with the results by Bet et al. (1996). The strong damping of the wave field in the reference results is due to rather coarse grid resolutions used in these cases. The insufficient grid resolution also shows as an increasing phase difference between the present and the reference results. Bet et al. used a structured grid with 50 nodes in the stream-wise and 30 nodes in the vertical direction. For the FINFLO simulation (Mikkola, 1999) a grid with twice as many points in both directions was used. However, due to a relatively diffusive boundary condition treatment in FINFLO which is similar to the Case a above, the level of damping in the reference results is roughly the same. In the present cases a grid with significantly higher resolution has been used. With 280 nodes the free surface discretisation is almost three and six times as high as in the reference cases. Because of these significant differences in the discretisation resolutions and in the level of the resulting discretisation error the present and the reference results are not comparable further away from the obstacle.

The differences in the flow field can most easily be seen from the contours of the v_2 velocity shown in Fig. 3.5 for the three different free surface approaches. Here the contours are drawn on the basis of values at the cell centres and averaged values at the grid points, including ghost cells. Cases b and c give relatively smooth and almost identical velocity distributions, whereas the velocity field from Case a has some oscillations close to the free surface. It can be seen that because of the zero gradient condition the contours mostly cross the free surface orthogonally, which clearly should not be the case. As a result of an incompatible boundary condition, there is thus a jump in the gradient field close to the free surface.

3.4 The results for the flow over a submerged hydrofoil

The second steady case focuses on the efficiency of the iteration with the different approaches. This was studied by simulating the case with two pressure under-relaxation factors (0.2 and 0.3) and with a set of five free surface pseudo time step parameters $C_{\Delta\tau f}$ ranging between 0.3 and 0.7. The range of free surface time steps was selected in order to quarantee convergence to machine accuracy with all of the approaches in each case. The bulk flow pseudo time step parameter $C_{\Delta\tau}$ was fixed at 1.0.

The results of the analysis of the convergence histories are presented in Tab. 3.2 for the first velocity component and the pressure. For brevity, the results for the second velocity component are not presented here. The full results can be found in Appendix A. On

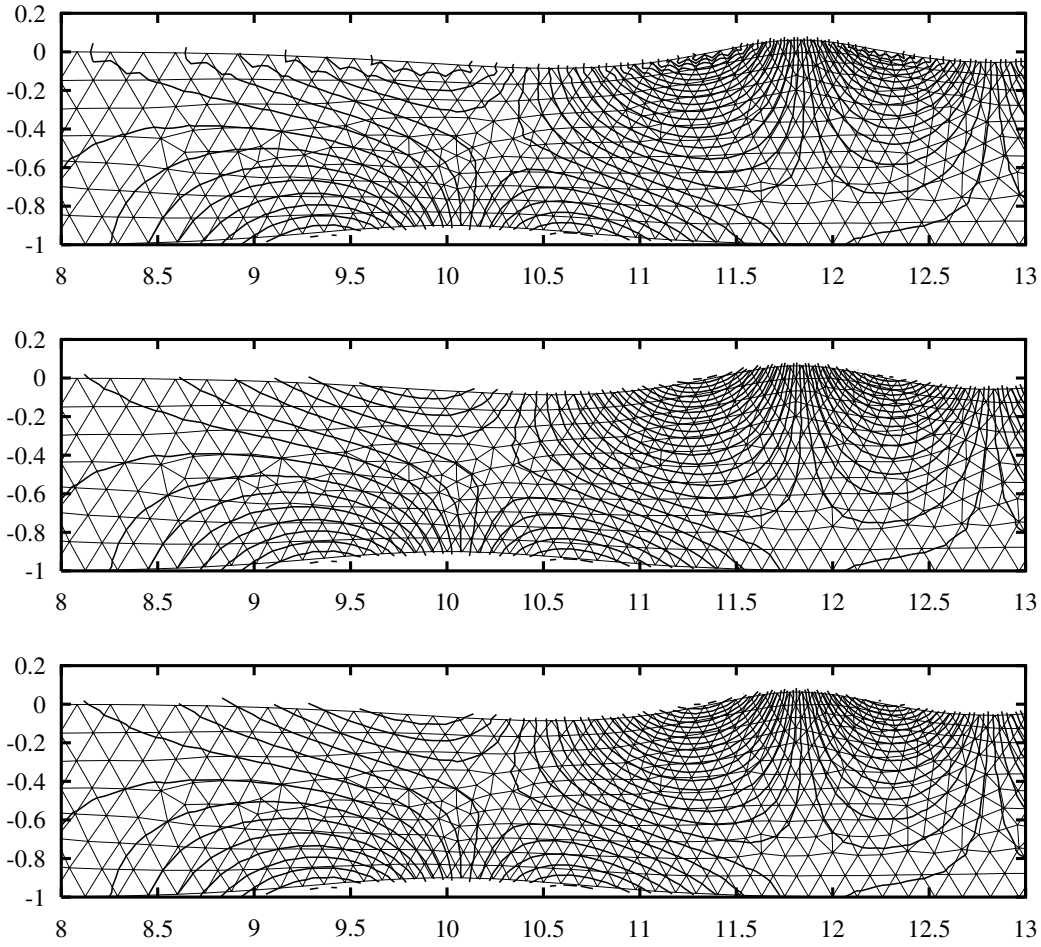


Figure 3.5: Contours of v_2 velocity for flow over a bump in Cases a, b, and c, respectively. The contour interval is 0.02.

Table 3.2: The results of the analysis of the convergence histories of the first velocity component and the pressure for the flow over a submerged hydrofoil. Top: the coupled kinematic approach (Case b); middle: the coupled dynamic approach (Case c); bottom: the decoupled kinematic approach (Case d).

	α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
Case b	0.2	0.3	.99832	9.0E-4	9.7E-8	.99834	7.9	5.3E-4
	0.2	0.4	.99798	9.9E-4	9.6E-8	.99800	8.8	5.2E-4
	0.2	0.5	.99764	1.0E-3	9.4E-8	.99767	9.0	4.9E-4
	0.2	0.6	.99726	1.2E-3	9.4E-8	.99730	10.	4.9E-4
	0.2	0.7	.99692	1.3E-3	9.5E-8	.99698	11.	5.0E-4
	0.3	0.3	.99797	8.0E-4	9.6E-8	.99799	4.8	5.3E-4
	0.3	0.4	.99757	7.3E-4	9.6E-8	.99761	4.3	5.4E-4
	0.3	0.5	.99708	9.3E-4	9.6E-8	.99714	5.3	5.3E-4
	0.3	0.6	.99654	1.3E-3	9.6E-8	.99664	6.9	5.3E-4
	0.3	0.7	.99586	2.2E-3	9.6E-8	.99603	11.	5.3E-4
Case c	0.2	0.3	.99876	8.5E-4	1.1E-7	.99878	6.0	5.0E-4
	0.2	0.4	.99842	1.0E-3	1.1E-7	.99843	7.8	5.0E-4
	0.2	0.5	.99814	1.1E-3	1.1E-7	.99812	9.8	4.9E-4
	0.2	0.6	.99789	1.3E-3	1.1E-7	.99788	11.	5.1E-4
	0.2	0.7	.99767	1.4E-3	1.1E-7	.99767	11.	4.9E-4
	0.3	0.3	.99830	1.0E-3	1.1E-7	.99830	5.5	5.3E-4
	0.3	0.4	.99787	1.3E-3	1.2E-7	.99788	7.0	5.3E-4
	0.3	0.5	.99748	1.6E-3	1.1E-7	.99750	8.4	5.3E-4
	0.3	0.6	.99719	1.7E-3	1.2E-7	.99722	8.6	5.4E-4
	0.3	0.7	.99692	1.8E-3	1.4E-7	.99694	9.4	5.5E-4
Case d	0.2	0.3	.99880	4.2E-4	9.4E-8	.99881	3.9	5.0E-4
	0.2	0.4	.99859	4.2E-4	9.5E-8	.99861	3.9	5.1E-4
	0.2	0.5	.99845	3.6E-4	9.5E-8	.99844	3.6	5.0E-4
	0.2	0.6	.99824	3.7E-4	9.4E-8	.99823	3.8	5.0E-4
	0.2	0.7	.99803	3.7E-4	9.4E-8	.99802	3.8	4.9E-4
	0.3	0.3	.99850	3.9E-4	9.6E-8	.99851	2.5	5.3E-4
	0.3	0.4	.99819	4.4E-4	9.6E-8	.99820	2.8	5.3E-4
	0.3	0.5	.99797	3.8E-4	9.6E-8	.99797	2.5	5.3E-4
	0.3	0.6	.99761	4.7E-4	9.7E-8	.99764	3.0	5.3E-4
	0.3	0.7	.99731	5.2E-4	9.6E-8	.99737	3.2	5.3E-4

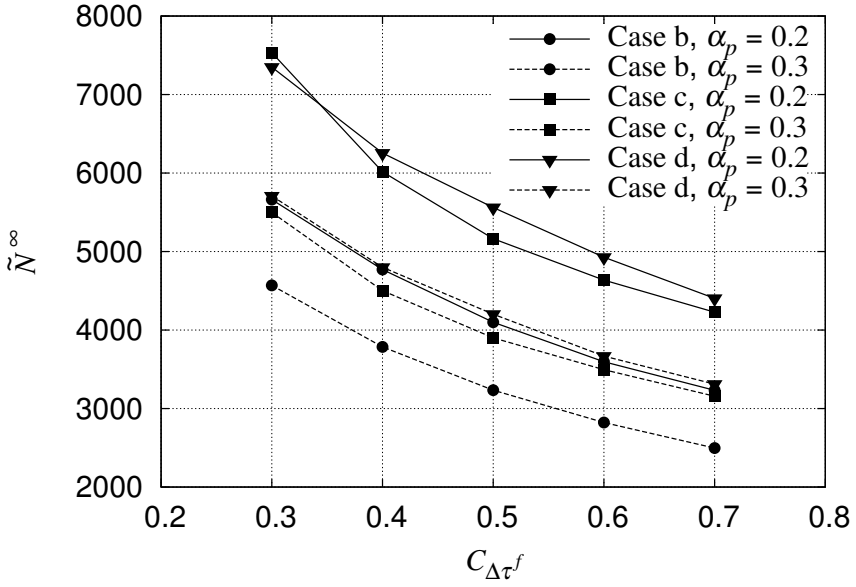


Figure 3.6: Comparison of the number of iterations required to get a converged solution with the different approaches for the flow over a submerged hydrofoil based on the approximate construction (3.2). Case b: the coupled kinematic approach; case c: the coupled dynamic approach; case d: the decoupled kinematic approach.

the basis of the analysed results a generic measure $\tilde{N_f}^{\infty}$ for the number of iterations required to reach converged solution was evaluated using Eq. (3.6) as explained in Sec. 3.2. Because each flow variable resulted in a slight different estimate, the average of the estimates for the velocity components and the pressure has been used for the comparison. The results are presented in Fig. 3.6.

The main conclusion from the results in Tab. 3.2 and Fig. 3.6 is that the coupling in the kinematic approach has a significant influence on the convergence rate. Even with the smaller pressure under-relaxation factor the convergence of the coupled approach (Case b) is slightly faster than the convergence of the decoupled counterpart (Case d) with the larger relaxation factor. With same under-relaxation factors the difference between the coupled and the decoupled approach is obvious. With the coupled dynamic approach (Case c) the convergence rate is comparable to the decoupled (Case d) rather than to the coupled (Case b) kinematic approach. It would thus seem that in terms of convergence rate the coupled kinematic approach is superior compared not only to the decoupled kinematic approach but also to the coupled dynamic approach.

The differences between the approaches are compared further by normalising the num-

ber of iterations for the coupled dynamic (Case c) and the decoupled kinematic approach (Case d) with the corresponding number of iterations for the coupled kinematic approach (Case b). This is shown in Fig. 3.7. It is observed that generally the relative difference in

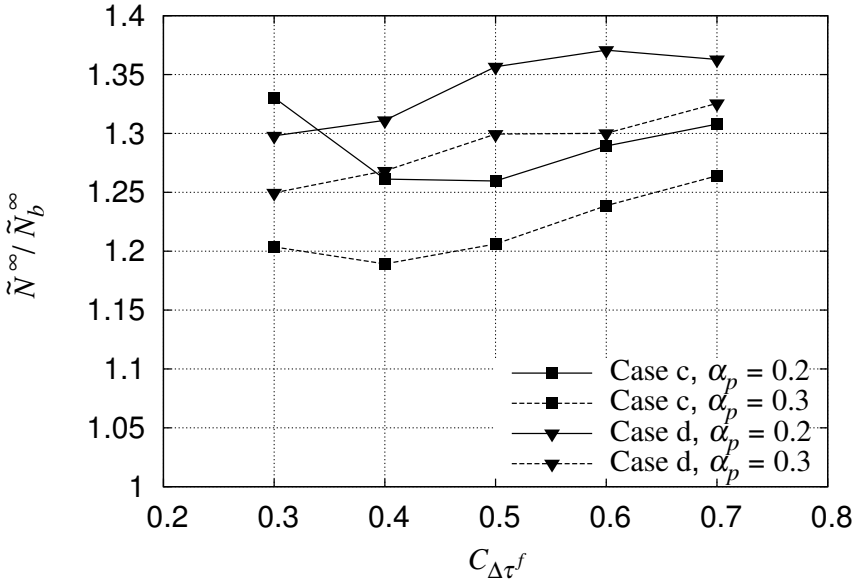


Figure 3.7: The relative difference in the required number of iterations compared to the coupled kinematic approach (Case b) for the flow over a submerged hydrofoil. Case c: the coupled dynamic approach; case d: the decoupled kinematic approach.

the convergence rate compared to the coupled kinematic approach increases with increasing free surface time step. However, with the coupled dynamic approach (Case c) the smallest time step results in increased differences particularly with the smaller pressure under-relaxation factor. This may be related to excessive under-relaxation of the pressure correction equation caused by the combination of the coupling (see Eq. 2.169) and the explicit under-relaxation.

The development of the wave field in the case of the kinematic (Case b) and the dynamic (Case c) approaches is compared in Fig. 3.8. The overall development of the wave elevation is very similar to the first test case. The kinematic approach (Case b) again exhibits faster initial development of the wave field. However, in this case there are some oscillations close to the damping zone border in the case of the dynamic approach (Case c), but they fade away during the iteration. A possible reason for the oscillations is the sudden growth in the cell size at the edge of the damping zone. Similarly to the first test case, the wave fields build up differently. In the kinematic approach the wave field compresses

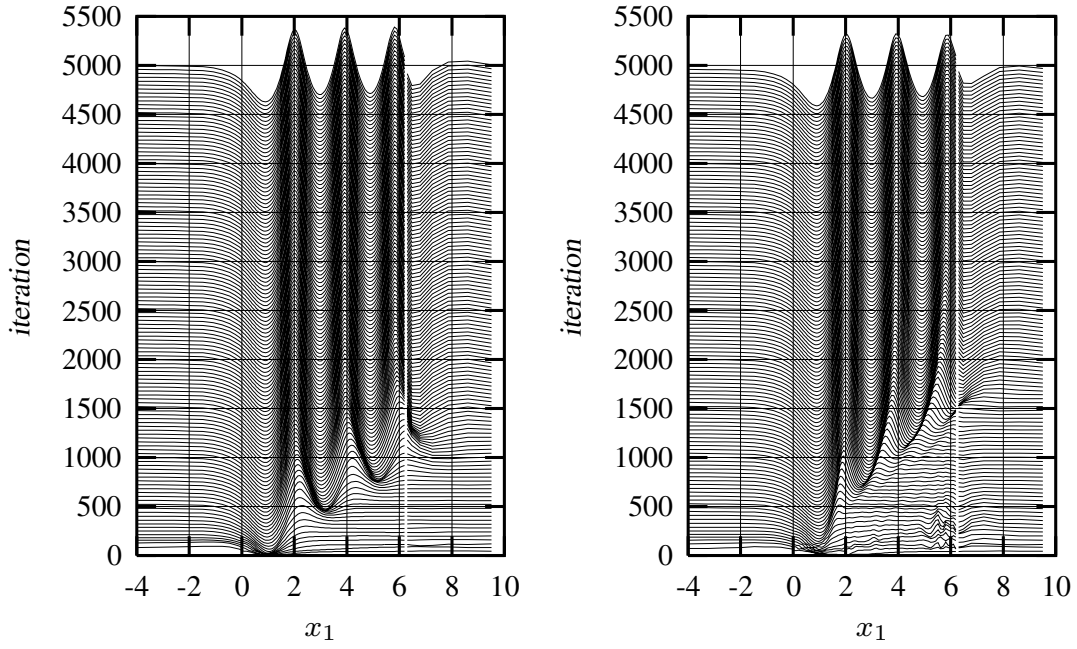


Figure 3.8: The convergence of the wave elevation in Cases b (kinematic, left) and c (dynamic, right) for the flow over a submerged hydrofoil ($\alpha_p = 0.2$, $C_{\Delta\tau f} = 0.5$).

towards upstream as the solution progresses, whereas in the dynamic approach expansion towards downstream can be observed.

Wave profiles for different approaches are compared with experimental and numerical references in Fig. 3.9. Good agreement with the numerical and experimental results can be seen with similar damping to the first test case. Compared to the experiment, the depth of the first through is, however, underestimated. This may be due to an insufficient resolution of the grid in front of the through, where the cell size can be seen to increase quite rapidly. There is also a small difference in the wavelength between the computations and the measurement.

Based on the results of this study and the study on the flow over a bump in the previous section the coupled kinematic approach has been chosen for the additional free surface studies, i.e. the following unsteady convergence study and the code verification study. The coupled kinematic approach has been observed to have superior convergence rate as well as better stability compared to the dynamic counterpart. Even if it was demonstrated that the coupling in the dynamic approch is effective, the convergence rate is mostly on a par with the decoupled kinematic approach. Furthermore, the stability of the dynamic approach seems to be overly sensitive to the characteristics of the flow case.

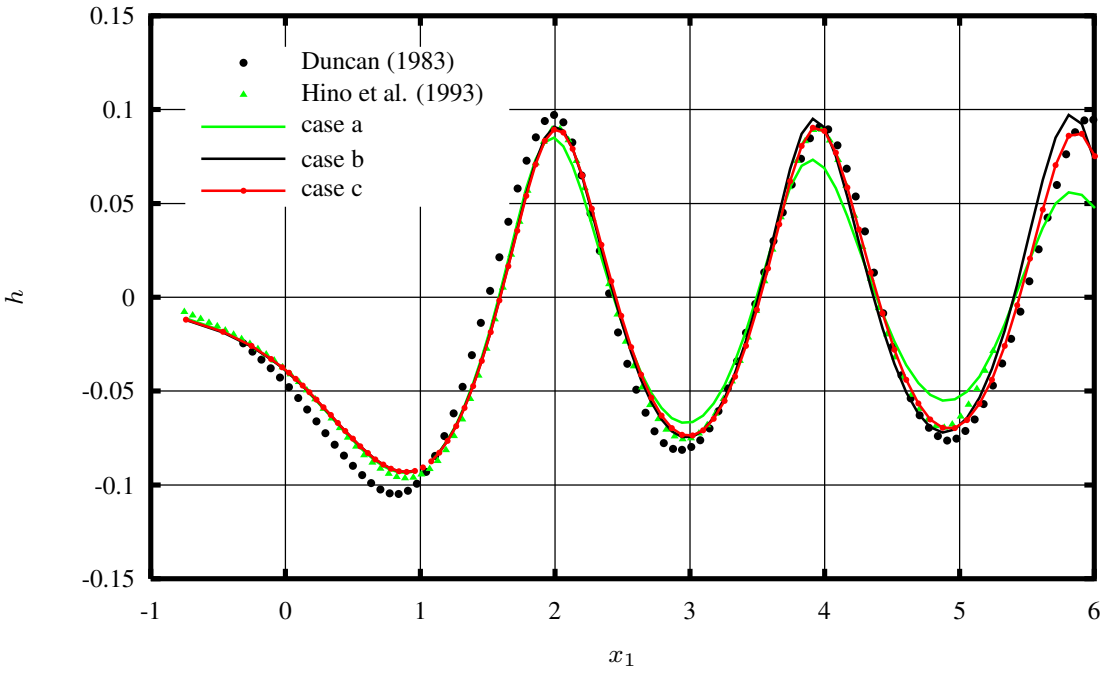


Figure 3.9: Wave profiles for flow over a submerged hydrofoil compared with measurements (Duncan, 1983) and numerical reference data (Hino et al., 1993). Case a: the kinematic approach with zero gradient extrapolation; case b: the kinematic approach with linear extrapolation; case c: the dynamic approach.

3.5 The results for a standing wave

As the simulation of unsteady flows is implemented with a steady approach for each physical time step, it is believed that the coupling has a similar influence also in unsteady cases. The influence may be less significant though, as the physical time derivative increases the diagonal terms of the systems of equations and, thus, induces under-relaxation to the solution process (see e.g. Eqs. 2.96 and 2.116). As the contribution to the diagonal term is inversely proportional to the physical time step, it is assumed that the possible influence of the coupling in unsteady cases is more significant in case of large physical time steps.

The influence has been studied by performing the simulations with the coupled and the fully decoupled kinematic approach (Cases b and d) using three alternative free surface pseudo time step CFL numbers, i.e. $C_{\Delta\tau^f} = 1.0$, 2.0 and 3.0. The bulk flow time step parameter $C_{\Delta\tau}$ was 5.0 in all three cases. The coupled and the fully decoupled approach have been compared in terms of the maximum changes of the solution variables for each

physical time step after 50 pseudo time steps. With the coupled approach the number of iterations is nearly sufficient to reach the machine accuracy with all three physical time steps.

The results are shown for the convergence of the first velocity component and the free surface deformation in Fig. 3.10. The results confirm the assumption that the influence of the coupling increases with incresing physical time step. For the smallest of the physical time steps tested the convergence of the coupled and fully decoupled approaches is very similar. As the time step is increased, differences start to emerge. For the medium time step the convergence speed decreases with the fully decoupled approach as the pseudo time step is decreased. The opposite is true for the largest physical time step. Particularly with the largest physical time step the convergence of the fully decoupled approach is very sensitive to the choice of the pseudo time step parameter. Additional testing has revealed that optimal convergence with the fully decoupled approach and the largest time step is reached using $C_{\Delta\tau^f} \approx 1.2$. On the other hand, the coupled approach seems to be extremely insensitive to the choice within the tested range. It should be emphasised that this finding applies only to the three tested pseudo time steps. It is obvious that, if the free surface pseudo time step is reduced significantly, the convergence of the coupled approach will decrease accordingly, as the evaluation of the surface evolution is based on time integration.

If the largest physical time step is used, the fully decoupled approach becomes unstable with a pseudo time step parameter of roughly $C_{\Delta\tau^f} = 4$. With the same time step, the coupled approach has been observed to be stable with any choice of the free surface pseudo time step, and even a value of $C_{\Delta\tau^f} = 10^{30}$ has been tested. However, with even larger physical time steps a stability limit for the pseudo time step has been observed for the coupled approach as well.

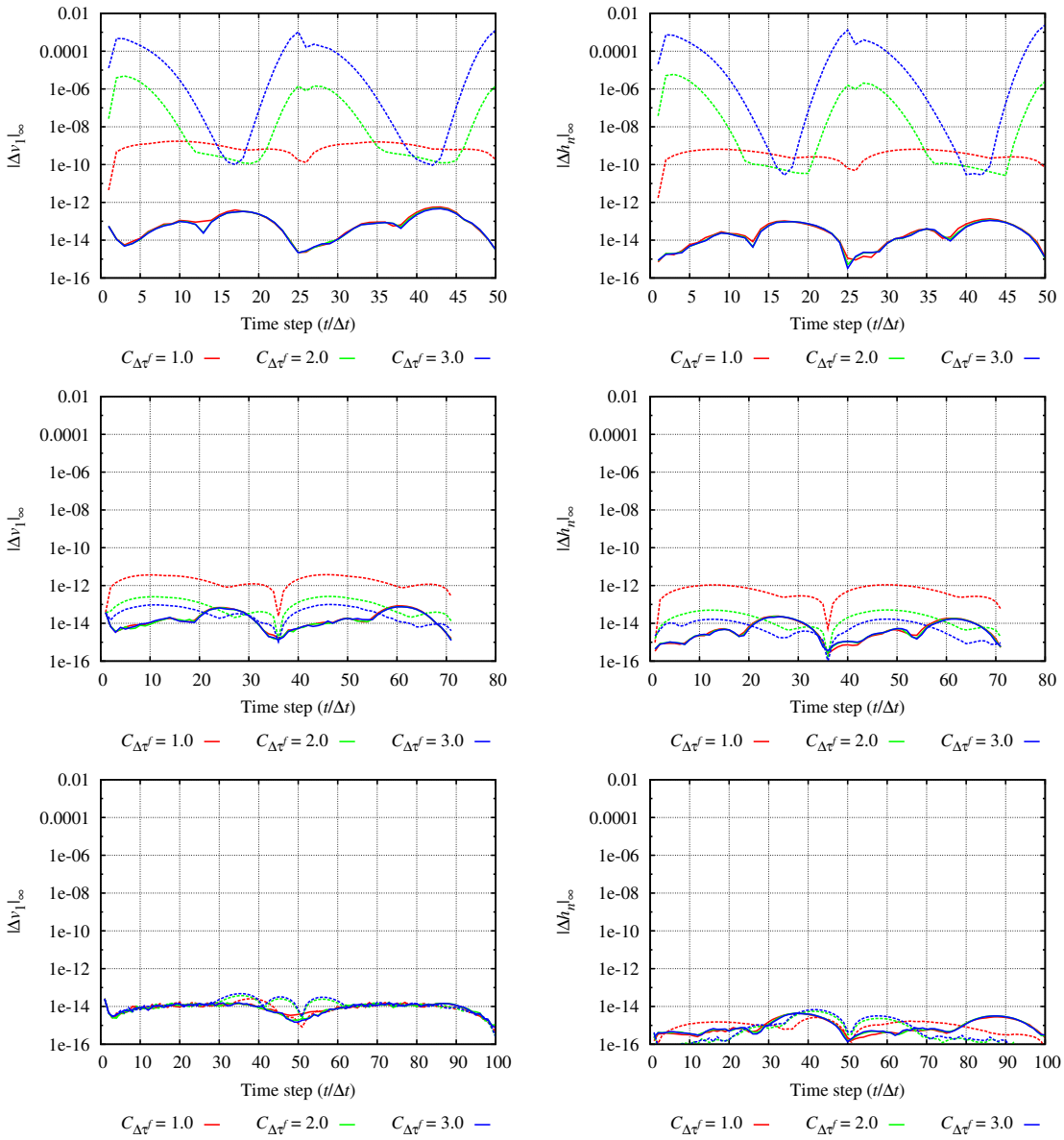


Figure 3.10: The maximum change of the first velocity component (left) and the free surface deformation (right) for each physical time step after 50 pseudo time steps in the standing wave test case. The number of physical time steps per one oscillation period is: 50 (top), 71 (middle) and 100 (bottom). Solid line: coupled kinematic approach (Case b); dashed line: decoupled kinematic approach (Case d).

Part II

Code verification

4 Code verification with the method of manufactured solutions

In Chapter 2 a theoretical presentation of the discretised approach for the solution of the governing equations was given. In order to have confidence that the conceptual model represented by the governing equations is implemented correctly, the developed code has to be verified. In this work the code verification was performed using the Method of Manufactured Solutions (MMS) (Steinberg and Roache, 1985; Roache, 1998, 2002; Salari and Knupp, 2000). Two code verification exercises with MMS were performed, one with a steady flow without a free surface and the other with an unsteady free surface flow.

This chapter consists of two sections: a section discussing the general aspects of numerical error and an introduction to MMS. The code verification exercises are presented in two separate chapters. These are followed by a chapter discussing the results of the code verification exercises.

4.1 Numerical error

Oberkampf and Trucano (2002) and Oberkampf et al. (2004) identify five primary sources of errors for a numerical procedure which is stable, consistent, and robust: insufficient spatial discretisation, insufficient temporal discretisation, insufficient iterative convergence, computer round-off, and computer programming.

The first two errors, associated with insufficient discretisation, are related to the discrete formulation of the computational model. For finite volume methods the error is induced by the approximation of the volume and surface integrals of the governing continuum equations with the discrete operators. The numerical solution which satisfies the discretised forms of the governing equations is generally not the same as the exact solution satisfying the analytical continuum equations. However, for a consistent method the discretisation error, i.e. the difference between the numerical ϕ and the exact solution ϕ_e, approaches zero as the representative discretisation interval is reduced to zero.

The discretisation error can be written with a series representation

$$\phi - \phi_e = g_p h^p + g_{p+1} h^{p+1} + g_{p+2} h^{p+2} + \cdots \ , \tag{4.1}$$

where p is the order of accuracy of the method. It is assumed that the coefficient functions g depend on the discrete operators used and on the derivatives of the exact solution ϕ_e, but not on the representative discretisation parameter h. The order of accuracy of the discretisation is dictated by the rate at which the error approaches zero as the discretisation is refined. When the discretisation parameter is sufficiently close to zero, the solution is within the asymptotic range. In this case the error is dominated by the lowest-order term, i.e.

$$\phi - \phi_e \approx g_p h^p \ , \quad \text{if } h \ll 1 \ . \tag{4.2}$$

The convergence rate is, thus, determined by the lowest-order term in Eq. (4.1).

The behaviour of the discretisation error related to particular discrete operators and a particular equation can be studied by constructing a modified equation which is a continuous representation of the discretised problem. The modified equation is based on Taylor series expansion of the discrete operators and on the substitution of these expansions into the discretised equations. The additional derivatives introduced in the process result in either phase or amplitude error (Lomax et al., 2001).

The error stemming from insufficient iterative convergence is related to the nature of the conceptual model. Because of the non-linearity of the governing equations and, possibly, of the boundary conditions, the solution of the problem has to be based on an iterative approach. Additionally, the linear systems resulting from the linearised discretisation of the governing equations are often solved with an iterative solver rather than with a direct method. With the method developed in this work, the need for an iterative approach is related to the non-linear convection term in the momentum equations, the segregated pressure correction concept, and the non-linear free surface boundary conditions. The related iterative solution process was described in Secs. 2.5.3 and 2.6 and was demonstrated in Chapter 3. If the iterative convergence is insufficient, the computational solution does not satisfy the discretised equations exactly. The difference between this and the exact solution of the discrete equations is called iterative error. A method for the estimation of the iterative error has been proposed by Eça and Hoekstra (2006b). The method is based on the analysis of the convergence histories and on the assumption of geometric progression of the solution changes. However, the quantification of the iterative error has been left outside the scope of this work by generally requiring sufficient convergence in each case.

The exact solution to the discrete equations referred to in the previous paragraph assumes that numbers can be presented with infinite precision. However, with computers numbers have a finite precision, i.e. there is only a finite set of numbers available, with which the exact numbers are approximated. This induces an additional error, called the round-off error. The amount of round-off error can be reduced by increasing the precision of the numbers. Eça and Hoekstra (2006b) state that in general the use of 64-bit floating point numbers with a precision of 15 digits is enough to guarantee that the round-off error is negligible compared to the other sources of numerical error.

Code verification, as defined in this work (see Chapter 1), deals with the demonstration that the errors associated with the discrete formulation, i.e. errors associated with the first two sources, approach zero as the discretisation is refined. This demonstrates that the method, as implemented, is consistent. The requirement for consistency does not rule out programming mistakes that may affect the order of convergence of the method. For example, a second-order method, if incorrectly implemented, may have first-order convergence, but would still be consistent. Thus, a more rigorous verification, advocated by e.g. Salari and Knupp (2000), can be performed by additionally studying the order of convergence and comparing it to the theoretical order of accuracy of the discretisations.

As the observable numerical error is the combination of errors from all the five sources, successful demonstration requires the errors associated with the other three sources to be negligible in comparison. In other words, the iterations have to be sufficiently converged, the round-off error has to be negligible, and the code has to be free of programming mistakes affecting the numerical accuracy of the method. According to Roy (2005), the iterative and round-off errors should be at least 100 times smaller than the discretisation error in order to ensure that these do not adversely impact the evaluation of the order of accuracy. A similar finding was made by Eça and Hoekstra (2006b), stating that as a rule of thumbs the influence of the iterative error on the numerical error is negligible, if it is two to three orders of magnitude below the discretisation error.

In the code verification studies of this work the following steps were taken in order to guarantee the negligibility of the iterative and round-off errors. The influence of the iterative error was minimised by converging the steady solutions, and in unsteady cases the solutions for each physical time step, to machine accuracy. In practice this means that the maximum change in all variables between successive iterations was reduced to machine zero, i.e. to as low a level as the precision of the floating point numbers and floating point operations permit. The associated remaining round-off error was minimised

by using 64-bit precision for all real numbers.

4.2 The method of manufactured solutions (MMS)

Salari and Knupp (2000) identify three limitations with the use of exact, analytical solutions of the governing equations for code verification, i.e. with the method of exact solutions (MES). They consider the major limitation of MES to be the difficulty, if not the impossibility, of creating a comprehensive suite of tests that adequately exercise the fully general governing equations. This is because the solution should exercise all the terms being tested in the equations. For example, solutions for the Poiseuille or Couette flows are not sufficient, as they do not activate the advection terms in the equations.

A secondary limitation of MES is related to the implementation of the exact solutions. Even if exact analytical solutions exist for the governing continuum equations, these are commonly represented by infinite series, complex integrals, and asymptotic series. For code verification, the value of the exact solution – to be compared with the value of the numerical solution – has to be evaluated at several points in space and time, which are dictated by the spatial and temporal discretisations. However, for the aforementioned types of exact solutions, the accurate evaluation of the value may be very difficult because of problems e.g. with the convergence of infinite series, with the numerical integration, or with singularities in the domain of integration. Another limitation with MES is the possibility of singularities in the exact solution itself, preventing order of accuracy studies and possibly even making the demonstration of consistency very difficult.

The limitations related to the use of exact analytical solutions for code verification can be avoided with MMS. In MMS, instead of trying to find the solution of the governing equations, one first comes up with, i.e. manufactures, an analytical solution and alters the governing equations in such a way that the manufactured solution satisfies the altered equations. This apparently complex procedure is in fact straightforward and simple to perform, wherein lies the elegance of the method.

The necessary steps in MMS are briefly described next. For a more thorough explanation with some examples, the reader is referred to e.g. (Roache, 2002). In MMS one starts by taking a suitable analytical solution, i.e. the manufactured solution, and substituting it into the original governing continuum equations. If the solution does not satisfy the equations, a residual is left over from the substitution. The altered equations are produced

by substituting a source term equal to this residual into the original equations. Assuming that the governing equation is

$$f(\phi) = 0 \tag{4.3}$$

the altered equation is then

$$f(\phi) - f(\phi_{ms}) = f(\phi) - Q = 0 \,, \tag{4.4}$$

where ϕ_{ms} is the manufactured solution and $Q = f(\phi_{ms})$ is the source term. The manufactured solution is now the solution of the altered equation as the source term cancels the residual of the original equation.

The boundary conditions are provided by the manufactured solution or the boundary conditions applied should be such that the manufactured solution satisfies them. For example, the boundary values for the inflow boundary conditions are given by the manufactured solution. On the other hand, if the manufactured solution has a symmetry property, a mirror boundary condition can be applied along the symmetry boundary.

Salari and Knupp (2000) and Roache (2002) present some remarks on the choice of the manufactured solution. First, the solution should not be trivial. It should be composed of smooth analytical functions and be general enough to exercise every term in the governing equations. Roache further adds that one wants a solution which also exercises all ordered derivatives in the error expansion. Relatedly, Salari and Knupp recommend that the solution derivatives are bounded by a small constant. This ensures that asymptotic range is reached with practical grid sizes. This recommendation also rules out singularities in the solution.

The use of MMS with a particular flow solver requires the solver to support arbitrary source terms. The alterations in the governing continuum equations, i.e. just the source terms, have to be implemented into the solver in question. As the method developed in this work is based on the finite volume approach, the source terms are given as volume integrals of analytical functions. In the discrete formulation these are evaluated with the same approximation as other volume integrals, i.e. as

$$Q_l = \int_l q dV = V_l q_l \,, \tag{4.5}$$

where q_l is the value of the analytical source term function at the cell centre of volume l.

The solution of the discretised form of the altered equations gives a numerical approximation of the manufactured solution. The accuracy of the approximation depends on the

discretisations of the equations and on the corresponding discretisation parameters, such as the cell size. Code verification can now be based on the comparison of the numerical approximation and the known analytical solution. As the discretisation parameters are reduced the numerical solution should approach the analytical continuum solution, giving proof of the consistency of the method. A more thorough code verification can be performed by further estimating the order of accuracy with a systematic parametric refinement, as first suggested by Steinberg and Roache (1985).

5 The verification of the steady bulk flow solution

The code verification was started by studying the grid convergence of the method that was developed with a steady flow without a free surface. The purpose of this study was to verify the implementation of the spatial discretisations in the momentum and continuity equations and to make observations on the differences between the implemented discretisations and on the influence of the grid refinement strategy.

5.1 The manufactured solution and the source terms

The governing equations for the flow are the incompressible Euler and continuity equations (2.7) and (2.4). For the evaluation of the analytical source terms of the altered equations the original equations in the differential forms are used. In Cartesian coordinates, using u and v for the velocity components and x and y for the coordinates, the finite volume formulation results in the steady, partial differential equations

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + u\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{5.1}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + v\frac{\partial u}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} \tag{5.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0\ . \tag{5.3}$$

Here the Gauss theorem has been used for the convection and pressure terms, as well as for the continuity condition, before the integral equations are transformed into differential equations. It should be noted that with the unaltered equations (5.1)-(5.3) for incompressible flow without mass sources the last two terms on the left-hand sides of the momentum equations (5.1) and (5.2) disappear because of the continuity condition (5.3). However, as the manufactured solution can be arbitrary, in a general case it does not satisfy the continuity condition (5.3), and the last two terms do not disappear. Therefore, when the source terms of the momentum equations are being evaluated, these terms cannot generally be dropped.

The general manufactured solution proposed by Salari and Knupp (2000) is used for the verification of the spatial bulk flow discretisation. Here the velocities and pressure are

given by

$$u_{ms}(x,y) = u_0 \left[\sin(x^2+y^2)+\varepsilon\right] \tag{5.4}$$

$$v_{ms}(x,y) = v_0 \left[\cos(x^2+y^2)+\varepsilon\right] \tag{5.5}$$

$$p_{ms}(x,y) = p_0 \left[\sin(x^2+y^2)+2\right] \tag{5.6}$$

For the sake of simplicity and without loss of generality, in this work $\varepsilon = 0$, $u_0 = 1$, $v_0 = 1$ and $p_0 = 1$.

The substitution of the manufactured solution (5.4)-(5.6) into the governing momentum and continuity equations (5.1)-(5.3) gives the source terms

$$\begin{aligned} q_u = \frac{2}{\rho}\big\{\rho\left[y\cos(2x^2+2y^2)+x\sin(2x^2+2y^2)\right] \\ + x\cos(x^2+y^2)\big\} \end{aligned} \tag{5.7}$$

$$\begin{aligned} q_v = \frac{2}{\rho}\big\{\rho\left[x\cos(2x^2+2y^2)-y\sin(2x^2+2y^2)\right] \\ + y\cos(x^2+y^2)\big\} \end{aligned} \tag{5.8}$$

$$q_m = 2x\cos(x^2+y^2) - 2y\sin(x^2+y^2) \tag{5.9}$$

for the momentum and continuity equations, respectively. The source terms have been substituted into the explicit residual (2.120) and into the mass balance error on the right-hand side of the pressure correction equation (2.150) using the discrete approximation for the volume integral (4.5).

5.2 The simulation case

For the verification exercise a rectangular domain was chosen. The computational domain is the same as the one used by Salari and Knupp and spans from $x = -0.1$ to $x = 0.7$ and from $y = 0.2$ to $y = 0.8$. The manufactured solution and the domain of interest are shown in Fig. 5.1.

The boundary conditions applied are based on the manufactured solution. As there is inflow at the left-hand and bottom boundaries, inflow-type conditions are applied at these boundaries, whereas outflow-type conditions are used at the right-hand and top boundaries. With inflow-type boundary conditions the velocity components are taken from the manufactured solution and the pressure is extrapolated to the boundary from the computational domain. With outflow-type conditions the opposite is done, i.e. the pressure is fixed

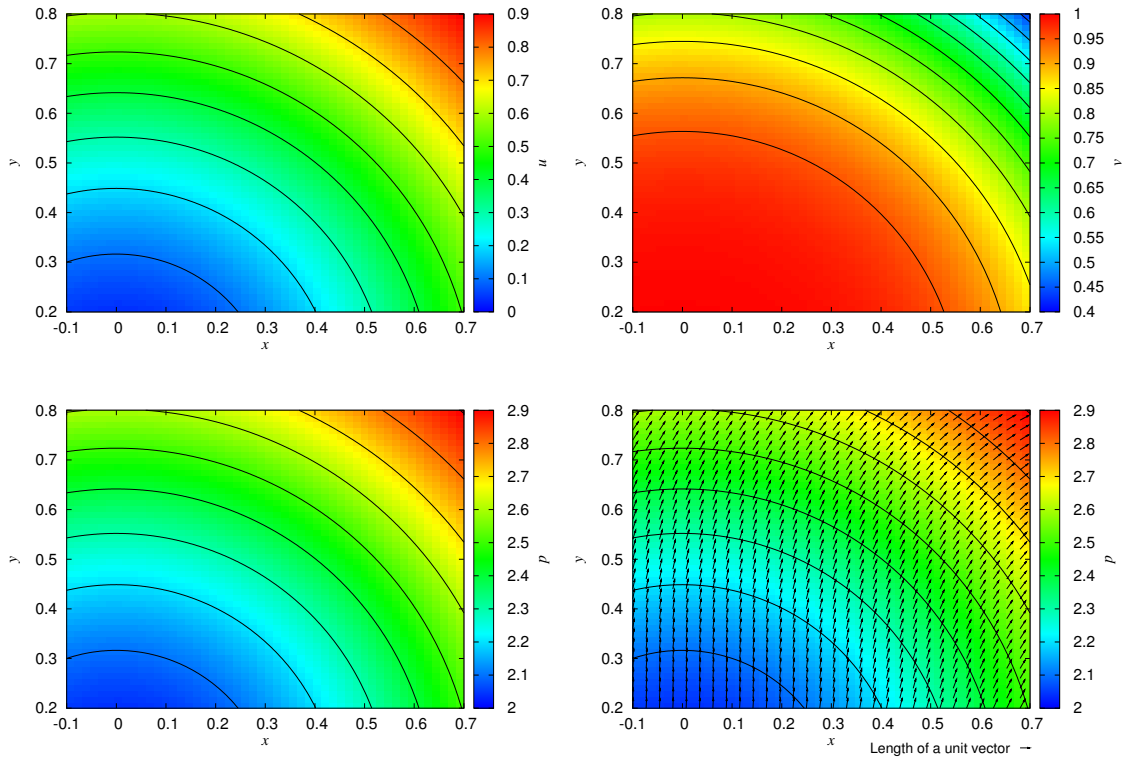


Figure 5.1: The manufactured solution and the computational domain for the bulk flow verification. Above: velocity components, below: pressure (left) and flow field (right). Contour intervals are 0.1 for u and p, 0.05 for v.

on the basis of the manufactured solution and the velocity components are extrapolated. The fixed values at the boundaries are enforced using Eq. (2.170). For the extrapolation both Eq. (2.172) and Eq. (2.173) were tested.

Similarly to the domain, the number of points on the boundaries is roughly the same as that in (Salari and Knupp, 2000). However, the corresponding number of elements inside the domain in the current work is significantly higher than in their study, as they used structured grids of quadrilaterals, whereas triangles were used here.

In order to study the influence of the technique used to refine the unstructured grids, two sets of triangular grids were used in the verification exercise. In the first set – referred to as Set A below – each successively refined grid was generated separately with a Delaundo grid generator (Müller, 1996). Delaundo generates unstructured triangular grids using the frontal Delaunay method. The input for Delaundo consists of the grid point distribution at the boundaries – defining the boundary geometry at the same time – and some parameters controlling the grid generation process. The control parameters influence, among others, the variation in the size and shape of the triangles generated. In Set A, the refinement was applied for the boundary point distributions. In order to maximise the geometric similarity of the grids, the same control parameters were used for every grid in the set.

In the second approach – used to generate the grids in Set B – the refined grids are generated sequentially from a previously generated, coarser grid by bisecting each edge of the grid and dividing each triangle into four triangles with the same shape. This same technique was used by Murali and Burg (e.g. 2002). In the literature this process is called primal triangle quadrisection (see e.g. Schröder, 2002). Hereafter, the technique is referred to for the sake of brevity as quadrisection refinement.

With both refinement techniques a special pre-processing step is required at the corners of the grid. It is quite common for the grids generated with Delaundo for rectangular domains to have a triangle at a corner with sides at both boundaries. If boundary conditions involving extrapolation are used at both boundaries, this topology leads to an under-resolved gradient in the cell at the corner. Therefore, before proceeding with the flow solution, each grid is checked for such triangles and a diagonal swap is performed to remove the problematic triangles. Then a Laplace smoothing step (see Eq. 2.201) is applied in order to improve the quality of the modified grid close to the corners. An example of the grid modification process is shown in Fig. 5.2.

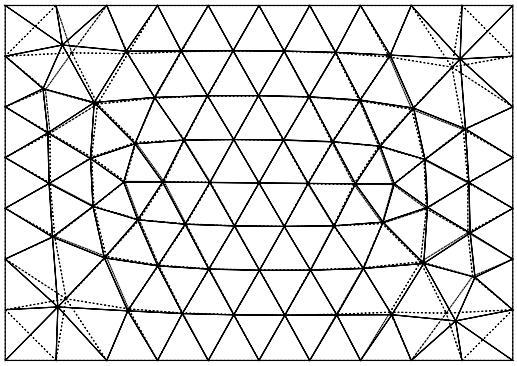


Figure 5.2: Removal of problematic triangles at the corners with diagonal swapping and Laplace smoothing. Solid line: modified grid, dashed line: original grid.

The Laplace smoothing applied has an added benefit in the case of quadrisection refinement. As, by definition, the quadrisection generates four identical triangles, the spacing distribution on the refined grid becomes piecewise constant with constant spacing within a coarse triangle and a jump at the border of coarse triangles. Such a spacing may influence the order of accuracy and, thus, may lead to erroneous conclusions in verification studies (Salari and Knupp, 2000).

Four and five levels of refinement were used with Sets A and B, respectively. Set A has only four levels of refinement as Delaundo failed to generate sufficiently small cells at the intersection of three advancing fronts if an even higher level of refinement was attempted. The unrefined grid (level 0) generated with Delaundo is the same in both sets. A summary of the properties of the grids is given in Tab. 5.1, where N_x is the number of faces at the bottom and top boundaries, N_y is the number of faces at the right-hand and left-hand boundaries, and N_e is the number of triangles in the grid.

It should be noted that Sets A and B have different numbers of faces at their boundaries. In Set B the number of faces at each boundary doubles with every level of refinement, because of the quadrisection process. In Set A the same is true for the bottom and top boundaries, but for the left-hand and right-hand boundaries the number of points at a boundary doubles instead. The first two levels of refinement for both sets are shown in Fig. 5.3. Whereas more isotropic grids are produced by refinement with Delaundo, a slightly better local geometric similarity between different levels is achieved with the quadrisection. This is partly caused by the non-uniform boundary refinement in Set A, as uniform refinement on the boundaries would increase local geometric similarity in this case too.

Table 5.1: The number of boundary faces and interior elements for different refinement levels.

Level		0	1	2	3	4	5
	N_x	10	20	40	80	160	-
Set A	N_y	7	15	31	63	127	-
	N_e	154	658	2782	11402	45978	-
	N_x	10	20	40	80	160	320
Set B	N_y	7	14	28	56	112	224
	N_e	154	616	2464	9856	39424	157696

The reason for the discrepancy at the boundaries in Set A is the inconsistent behaviour of the boundary geometry pre-processor for Delaundo. Nevertheless, the relative difference between the number of points and faces at a boundary decreases asymptotically with grid refinement. Therefore, as the level of refinement increases, the refinement with Delaundo becomes more uniform. Furthermore, the evaluation of the order of accuracy is based on the total number of elements in the domain rather than on the number of boundary faces, and as such the evaluation cannot differentiate between uniform and non-uniform refinement. It was also assumed that the inclusion of the set can reveal possible effects that a slightly dissimilar refinement may have on the grid convergence. Thus, the current sets of grids were used.

In addition to the study of the influence of the grid refinement technique on the verification results, the behaviour of the numerical error with different discretisations was also considered. Three discretisations were studied for the convected velocity component v_i in the convection part of the momentum flux (2.34): the first-order discretisation (2.47), the discretisation proposed by Frink (2.46), and the weighted Taylor extrapolation (2.45). Two approaches were used for the approximation of the pressure term (2.35) in the momentum flux. In the first approach the approximation is based on Gauss integration, (2.35) and (2.50), whereas in the second approach the pressure term is evaluated directly by using the volume integral (2.52) of the least-squares gradient approximation. The influence of the approximation of the gradients was studied by comparing the numerical error with gradients based on Gauss integration (2.76) and on the least-squares approach (2.81). For the Gauss integration-based evaluation of the gradients nine iterations of Eq. (2.76) were used. The effectiveness of the skewness correction was studied by performing the simulations with and without the correction. In the tested case this has an influence on the evaluation of the convection velocity $\bar{v}$ in the momentum flux and the mass flux (see

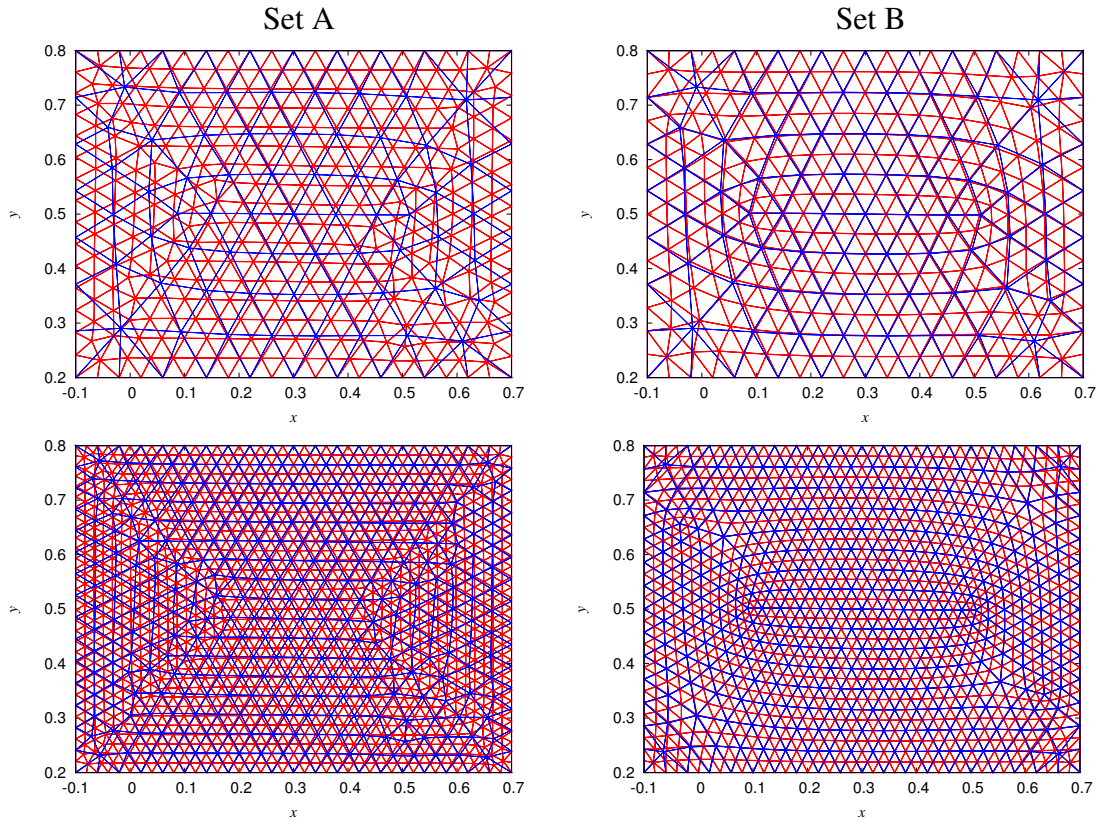


Figure 5.3: The influence of refinement technique on the resulting grids with the first two refinements. Above: levels 0 and 1, below: levels 1 and 2

Eqs. 2.43 and 2.157), as well as on the evaluation of the pressure in the momentum flux (see Eq. 2.50). As a final comparison, the influence of the discretisation of boundary conditions involving linear extrapolation was studied by using both the Taylor extrapolation-based approach (2.172) and the opposite node-based approach Eq. (2.173).

Tab. 5.2 gives a summary of the tested cases and comparisons with the set of grids and discretisations used.

5.3 The analysis methodology

In order to demonstrate consistency and order of accuracy of the method one has to study the behaviour of the difference between the numerical and the exact solution, i.e. the numerical error, as the discretisation is refined. In this exercise the global norm of the

Table 5.2: The tested cases for the steady manufactured solution. Taylor, Frink, and 1-st for the convection refer to Eqs. (2.45), (2.46) and (2.47), Gauss and Grad for the pressure refer to Eqs. (2.50) and (2.52), and Least and Gauss for the gradient refer to Eqs. (2.81) and (2.76), respectively. The skewness column indicates whether skewness correction was used. Grad and Oppos in the boundary condition (BC) column refer to the Taylor extrapolation based (2.172) and the opposite node-based (2.173) discretisations, respectively.

Comparison	Grids	Convection	Pressure	Gradient	Skewness	BC	Case
Grid	Set A	Taylor	Gauss	Least	Yes	Grad	1
	Set A	1-st	Gauss	Least	Yes	Grad	2
	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	1-st	Gauss	Least	Yes	Grad	4
Convection	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	Frink	Gauss	Least	Yes	Grad	5
	Set B	1-st	Gauss	Least	Yes	Grad	4
Pressure	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	Taylor	Grad	Least	Yes	Grad	6
Gradient	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	Taylor	Gauss	Gauss	Yes	Grad	7
Skewness	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	Taylor	Gauss	Least	No	Grad	8
BC	Set B	Taylor	Gauss	Least	Yes	Grad	3
	Set B	Taylor	Gauss	Least	Yes	Oppos	9

error over the domain was preferred to the local error for quantitative studies. The reasons for this are briefly discussed next.

The local error is affected by the local properties of the grid. With a general set of refined unstructured grids, significant non-systematic variation in the properties can occur between the grid levels. In such a case reliable evaluation of the order of accuracy based on local error is extremely difficult. Furthermore, a comparison of local errors on different grid levels requires the evaluation of the errors at the same locations. In general, however, points on different levels do not coincide and thus the results should be interpolated to a common location. This would lead to an additional interpolation error. There are some problems related to this. First, the interpolation error depends on the distance between the source and target points, whereas the discretisation error depends on the local grid properties. Second, the order of accuracy of the interpolation has to be higher than the order of the solution method to avoid an adverse impact on the observed order. In the case of unstructured grids, it is difficult to accomplish this. Nevertheless, local errors are used in this work to qualitatively study the behaviour of the numerical error and to study the influence of the local error on the global error norm.

In the present verification study a scaled L_2-norm of the numerical error was used to measure the global error. For a general variable ϕ this is defined as

$$err_2(\phi) = \frac{\sqrt{\sum_{l=1}^{N_e} \left(\phi_l - \phi_{ms}(x_l, y_l)\right)^2}}{\sqrt{N_e}} \,, \tag{5.10}$$

where ϕ_l and $\phi_{ms}(x_l, y_l)$ are the values of the numerical and exact solutions at the centre (x_l, y_l) of the finite volume l, respectively.

The evaluation of the order of accuracy was based on the one-term estimation of the error

$$err_2(\phi) = a\Delta\tilde{x}^p \,, \tag{5.11}$$

where $\Delta\tilde{x}$ is some general measure of the discretisation, p is the observed order of accuracy, and a is a constant. The inverse of the square root of the number of elements $1/\sqrt{N_e}$ was used as the discretisation measure, whereas p and a were obtained by fitting the function to the simulation results. As there are two free parameters in the function, results on at least two grid levels are required.

Two types of fits were performed. In the first fit results on the level i and the coarser level $i-1$ were used to determine the observed order of accuracy $p_\phi^{i,2}$ for variable ϕ. By taking

the logarithm on both sides of Eq. (5.11), subtracting the resulting equations for the levels $i-1$ and i, and rearranging the terms one gets

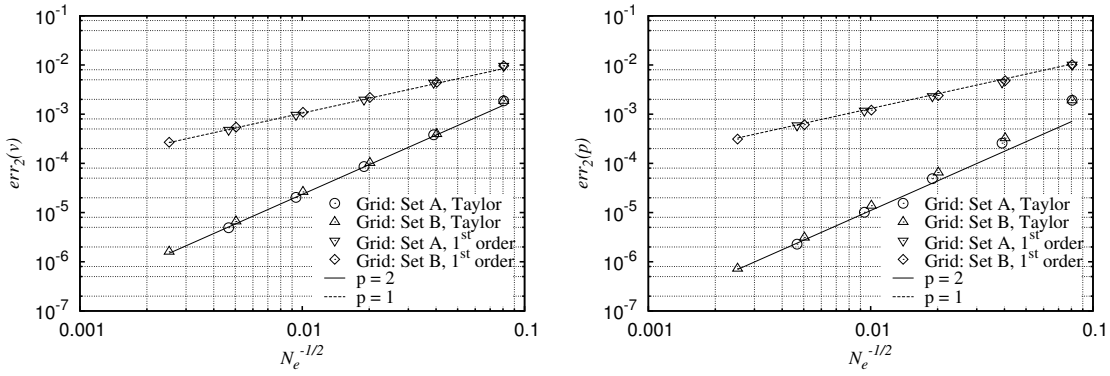
$$p_\phi^{i,2} = \frac{\ln|err_2(\phi^{i-1})| - \ln|err_2(\phi^i)|}{\ln\left(\Delta\tilde{x}^{i-1}/\Delta\tilde{x}^i\right)} \,. \tag{5.12}$$

In the second fit results on three levels from the levels i to $i-2$ were used. With just two free parameters the problem is over-determined and, thus, the parameters were solved using least -squares fitting. In this case both linear and non-linear fitting were applied. In the linear approach, similarly to what was done in the first fit, the logarithm was taken on both sides of Eq. (5.11), and a linear least-squares fit was performed on a log-log scale. In the non-linear case Eq. (5.11) was used directly in the fitting. The corresponding observed order of accuracy is denoted by $p_{\log\phi}^{i,3}$ and $p_\phi^{i,3}$ with the linear and non-linear fit, respectively. In both cases the actual fitting was performed using the fit option in gnuplot 4.2 (information and the program are available at http://www.gnuplot.info).

5.4 The results

The main results of the first verification study are discussed in this section. For the sake of clarity and easy access, the full results for each test case are collected in Appendix B. The grid convergence for the x-velocity is qualitatively very similar to the convergence of the pressure, and thus, for the sake of brevity, the graphs in this section present results only for the global error norm of the y-velocity and pressure.

The post-processing of the error maps was performed by colouring each control volume with a uniform colour corresponding to the value at the centre of the cell. This is the most faithful representation of the error and differs from the conventional, default practice of most post-processors. It is common that a visualisation software for unstructured data requires values at the vertices of the grid even in case of cell centred data. Thus, normally the cell centre data has to be averaged or interpolated to the vertices before visualisation. Even if the visualisation software accepts data in a cell centre format, the data is usually interpolated to the vertices internally by the software before visualisation.



	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
	4	1.97	2.04	2.14	2.01	2.04	2.18	2.04	2.05	2.22
Set A,	3	2.05	2.05	2.22	2.14	2.05	2.27	2.22	2.06	2.31
Taylor	2	2.23	2.06	2.31	2.42	2.12	2.54	2.58	2.18	2.74
	1	2.60	2.19	2.76	-	-	-	-	-	-
	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
Set B, Taylor	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
	4	1.01	1.01	0.98	1.01	1.01	0.97	1.01	1.01	0.97
Set A,	3	1.01	1.01	0.96	1.04	1.05	0.93	1.05	1.07	0.92
1^{st}	2	1.07	1.10	0.90	1.08	1.09	1.02	1.08	1.09	1.08
order	1	1.09	1.09	1.14	-	-	-	-	-	-
	5	1.01	1.02	0.97	1.01	1.01	0.98	1.00	1.01	0.98
Set B,	4	1.00	1.01	0.98	1.01	1.00	0.99	1.01	1.00	0.99
1^{st}	3	1.01	1.00	0.99	1.02	1.01	0.99	1.03	1.02	1.00
order	2	1.04	1.02	1.00	1.07	1.07	1.04	1.09	1.09	1.06
	1	1.11	1.12	1.09	-	-	-	-	-	-

Figure 5.4: Comparison of the influence of the grid set on the convergence of velocity and pressure.

5.4.1 The influence of the grid set

The results for the first comparison, i.e. for the influence of the grid set on the grid convergence and order of accuracy, are shown in Fig. 5.4. The results show that the observed order of accuracies based on the global error norm depends on the fit used. If the order of accuracy reduces monotonically with grid refinement, the higher order on coarser grids increases the order of the three-grid fits compared to the two-grid fit. With the non-linear fit this influence is greater than with the linear log-log fit and the orders observed are generally higher.

With the higher-order discretisation the order of accuracy reduces monotonically in all cases, except in the case of the y-velocity component with grid set B. The order of accuracy of the pressure converges towards two significantly more slowly than the orders of the velocities.

The orders well above two for the pressure with both grid sets are explained by the behaviour of the local error in the upper right-hand corner of the domain. With coarser grids this is also explained by the behaviour close to the bottom and left-hand boundaries (see Figs. B.4 and B.12). At these locations the error reduces significantly faster than with second-order convergence. The local error of the x-velocity component shows similar behaviour close to the upper left-hand corner and the left-hand boundary (see Figs. B.2 and B.10). For the y-velocity component the distribution of the local error already shows second-order convergence on rather coarse grids (see Figs. B.3 and B.11). Compared to the x-velocity component and the pressure, the local error in the upper right-hand corner is smaller and varies significantly less between grid levels.

In addition to the aforementioned areas of large local error, another interesting feature is the increased error caused by local irregularities in the grid properties, particularly for the velocities. With both grid sets an area of large local error originates from a point close to the lower left-hand corner and extends all the way to the upper boundary. With Set A there is a similar area to the right of the centre of the domain and with Set B close to the lower right-hand corner. A characteristic feature of all of these is the generation of large local errors as a result of a local irregularity and the convection of the error with the flow. The variation of the order of accuracy for the y-velocity component in Set B is most probably caused by the variation in the extent and strength of these localised errors, particularly by the variation in the one close to the left-hand boundary.

With the first-order discretisation the order of accuracy is very close to one, except with the coarsest grid pairs. Compared to the higher-order discretisation, the difference between the orders of accuracies for different variables is not as clear. However, with several grid pairs the order of accuracy for the pressure is below one.

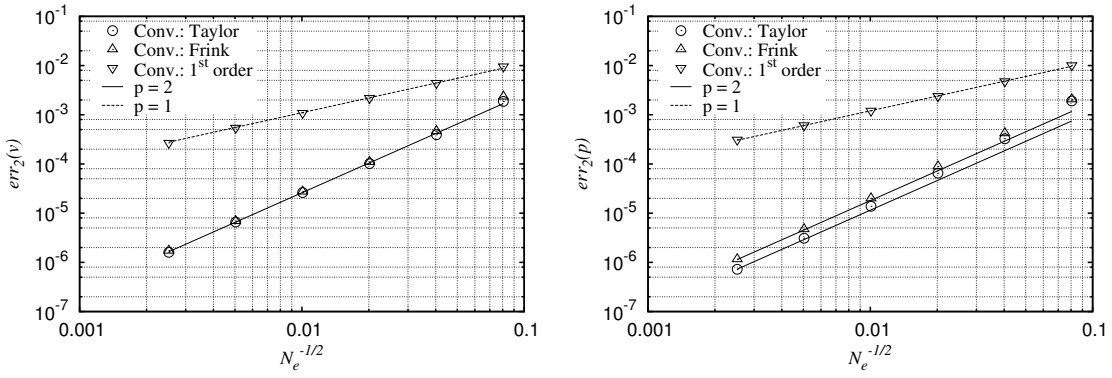
The order of accuracy close to one for the velocities in both sets is already obvious from the distribution of the local error (see Figs. B.6, B.7, B.14 and B.15). The distinctive areas of higher local error are already practically identical in shape and size from rather coarse grids on. The higher order of accuracy for the coarsest grid pairs is explained by the behaviour of the local error close to the right-hand boundary.

The local error distributions for pressure (see Figs. B.8 and B.16) show that for the coarsest grid pair of both sets the error in the upper and lower right-hand corners reduces faster than with first-order convergence. The resulting orders based on the global error norm are $p = 1.09$ and $p = 1.14$ for Sets A and B, respectively. However, for the next grid pair of Set A the error in the lower part of the domain is clearly slower than the first-order reduction (global order $p = 0.90$). With the grid pair of Set B, too, this area of large local error grows with grid refinement, but significantly less. With further refinement the variation in the error with Set A begins to stabilise and the order of accuracy of pressure approaches one, but with Set B the area of large error in the lower part extends slowly with refinement.

5.4.2 The influence of the convection discretisation

The influence of the discretisation of the convected velocity component is compared in Fig. 5.5. The results for the Taylor extrapolation-based discretisation and for the first-order discretisation have already been discussed to some extent, when the influence of the grid set was being compared, and thus the results for the Frink discretisation are considered below in more detail. The convergence of the observed order of accuracy towards two with the Frink discretisation is practically monotonous for all variables. With the finer grids the observed order of accuracy is closer to the theoretical order of two than with the simple Taylor discretisation.

In terms of local error of the velocity components, some essential differences are observed between the higher-order discretisations (see Figs. B.10 and B.18 and Figs. B.11 and B.19). The error in the velocity in the upper right-hand corner is clearly smaller



	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
Taylor	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
Frink	5	1.99	2.00	2.03	2.00	1.99	2.05	2.01	1.99	2.08
	4	2.01	1.99	2.08	2.04	2.00	2.12	2.07	2.00	2.15
	3	2.07	2.00	2.16	2.12	2.04	2.20	2.16	2.07	2.23
	2	2.17	2.08	2.24	2.29	2.21	2.26	2.40	2.32	2.28
	1	2.41	2.33	2.28	-	-	-	-	-	-
1st order	5	1.01	1.02	0.97	1.01	1.01	0.98	1.00	1.01	0.98
	4	1.00	1.01	0.98	1.01	1.00	0.99	1.01	1.00	0.99
	3	1.01	1.00	0.99	1.02	1.01	0.99	1.03	1.02	1.00
	2	1.04	1.02	1.00	1.07	1.07	1.04	1.09	1.09	1.06
	1	1.11	1.12	1.09	-	-	-	-	-	-

Figure 5.5: Comparison of the influence of the discretisation of the convected velocity component on the convergence of velocity and pressure.

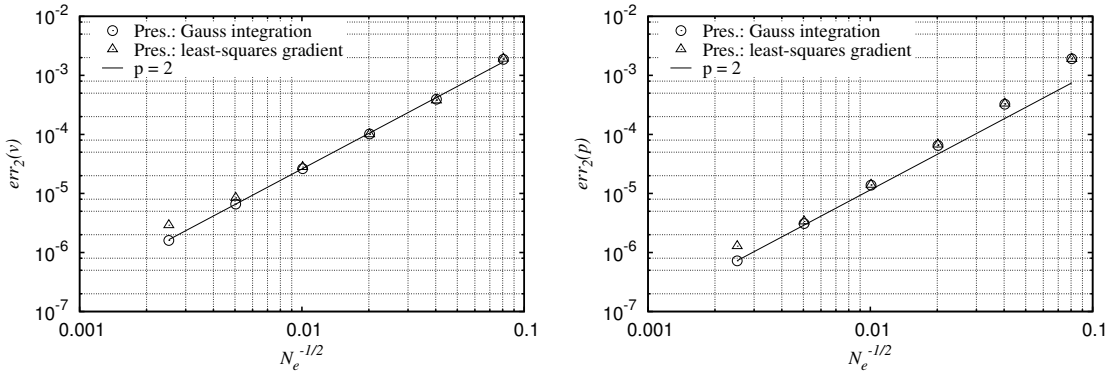
with the Frink discretisation, particularly for the x-component (see Figs. B.10 and B.18). For the y-velocity component another difference is the area of large error in the lower right-hand corner (see Figs. B.11 and B.19). On coarse grid levels the error compared to the rest of the domain is significantly higher with the Frink discretisation than with the Taylor discretisation. However, for both velocity components the error streak generated by the local grid irregularity in the lower right-hand corner of the domain is practically insignificant with the Frink discretisation. A similar observation can also be made for the error streak close to the left-hand boundary.

The distribution of local error of pressure is totally different between the higher-order discretisations (see Figs. B.12 and B.20). However, the orders of accuracy above two with the Frink discretisation are caused by the same phenomena as with the simple Taylor discretisation, i.e. by the higher order of the reduction of the error in the upper right-hand corner and close to the bottom and left-hand boundaries. With the finest grids the only major area of large local error with the Frink discretisation is in the upper left-hand corner.

With the first-order discretisation the level and the distributions of local error for all variables are totally different than with the higher-order discretisations (see Figs. B.14, B.15 and B.16). The observed order based on the global error norm is very close to the theoretical order of one. The influence of the local grid irregularities is significantly weaker with the first-order discretisations than with the higher-order ones, especially in the case of the velocities and particularly close to the lower left-hand corner. For the x-velocity component the largest errors are located around a region extending from the middle of the top boundary to the middle of the right-hand boundary, whereas for the y-velocity component the largest errors are located at the right-hand boundary and at the top boundary close to the upper left-hand corner. The largest errors in pressure are located in the lower part of the domain.

5.4.3 The influence of the pressure gradient discretisation

Fig. 5.6 compares the influence of the discretisation of the pressure term in the momentum flux on the order of accuracy of the method. Whereas the approach based on Gauss integration, already discussed with the previous comparisons, shows consistent behaviour with an order of accuracy close to the theoretical order, the approach based on the volume integral of the least-squares gradient shows a rapidly decreasing order of accuracy



	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
Gauss integration	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
Least-squares gradient	5	1.21	1.54	1.39	1.43	1.64	1.73	1.58	1.70	1.98
	4	1.66	1.73	2.07	1.83	1.79	2.16	1.96	1.83	2.25
	3	2.00	1.84	2.26	2.10	1.87	2.28	2.18	1.90	2.30
	2	2.20	1.91	2.31	2.32	2.13	2.41	2.43	2.31	2.50
	1	2.45	2.35	2.51	-	-	-	-	-	-

Figure 5.6: Comparison of the influence of the discretisation of the pressure gradient in the momentum equations on the convergence of velocity and pressure.

with grid refinement. The smoothest reduction is observed for the y-velocity component, whereas significant drops are observed for the x-velocity component and pressure when moving onto the finest grid.

For the coarser grids the behaviour of the local error is mostly similar with both methods for all variables (see Figs. B.10-B.12 and Figs. B.22-B.24). However, when the least-squares gradient-based approach is used, the area of large error in the upper right-hand corner starts to expand with grid refinement on the finest levels for both velocity components. Similarly, in the finest grid the streak of high error in y-velocity originating from the lower left-hand corner intensifies.

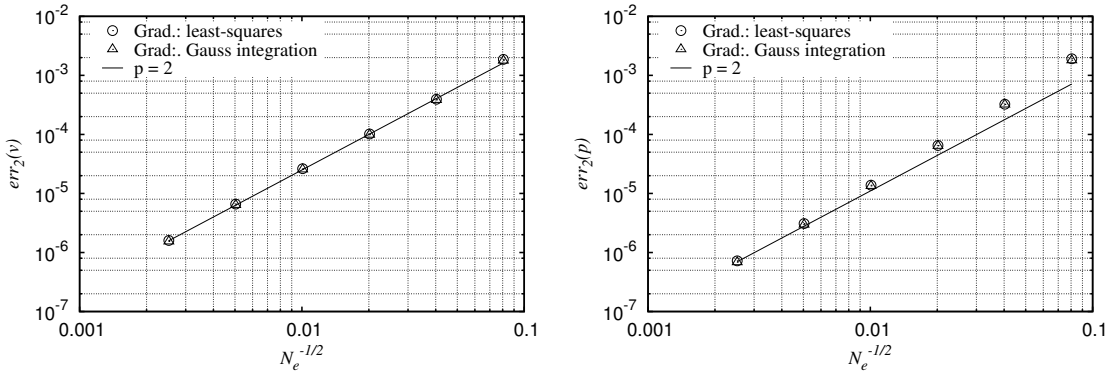
For the pressure the error right in the top right-hand corner and close to the bottom and left-hand boundaries reduces at a rate higher than two with both methods. However, differences between the methods are observed close to the upper left- and right-hand corners. With the Gauss integration, there is a large area of high local error in the upper left-hand corner, which does not exist with the least-squares gradient-based approach. On the other hand, with the latter approach an area of high local error that grows rapidly with the refinement of the grid is observed close to the upper right-hand corner. At the same time the error in the left half of the domain reduces with a higher-than-theoretical order until a rapid change in the order is observed on the finest grid.

5.4.4 The influence of the discretisation of the gradients

The influence of the discretisation of the gradients is compared in Fig. 5.7. The results show convincingly that both methods for gradient approximation perform equally well. The global error norms and the orders of accuracy are practically identical, except for the coarsest grid pair. This is a direct consequence of the local error distributions, for which no significant differences can be observed between the methods (see Figs. B.10-B.12 and Figs. B.26-B.28).

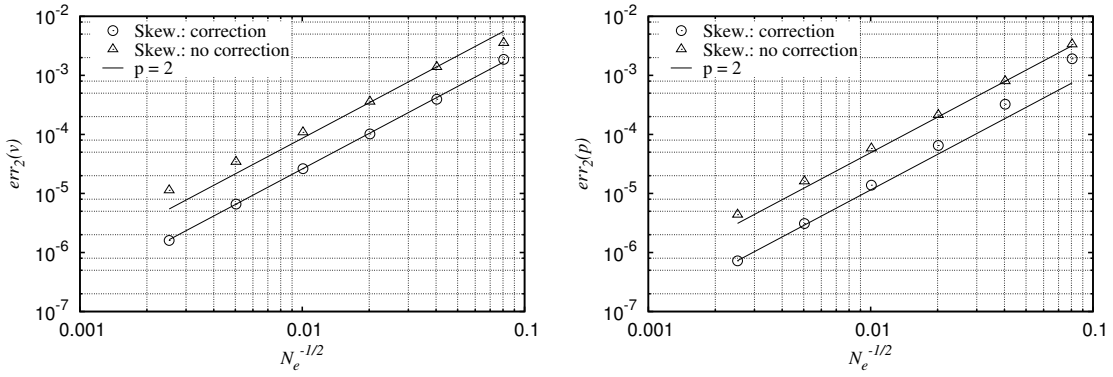
5.4.5 The influence of the skewness correction

Fig. 5.8 compares the results with and without the skewness correction. Without the skewness correction the order of accuracy for all variables is primarily below two. For the velocities the order reduces monotonically, except for the coarsest pair of grids.



	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
Least-squares	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
Gauss integration	5	2.06	2.05	2.12	2.06	2.03	2.15	2.07	2.00	2.17
	4	2.07	2.00	2.18	2.09	1.98	2.22	2.11	1.96	2.25
	3	2.11	1.95	2.25	2.15	1.96	2.29	2.18	1.96	2.32
	2	2.19	1.96	2.32	2.33	2.08	2.41	2.45	2.19	2.48
	1	2.46	2.21	2.49	-	-	-	-	-	-

Figure 5.7: Comparison of the influence of the discretisation of the gradients on the convergence of velocity and pressure.

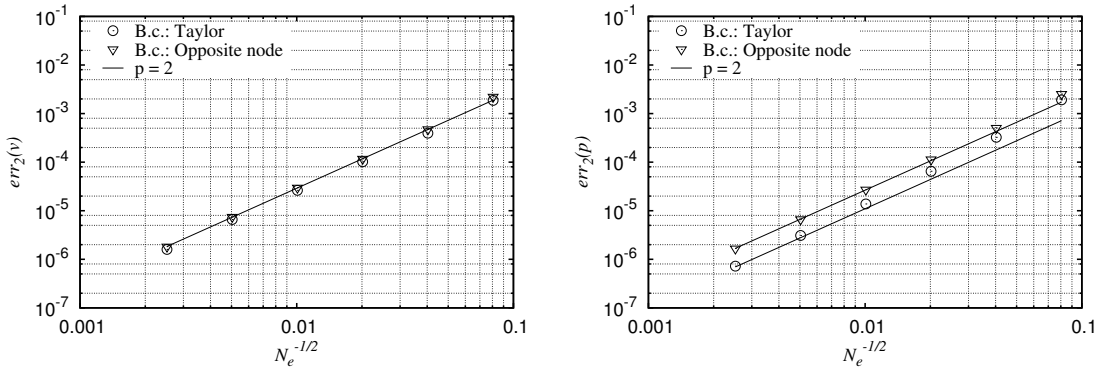


	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
Correction	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
No correction	5	1.77	1.60	1.88	1.79	1.63	1.86	1.81	1.65	1.85
	4	1.82	1.66	1.85	1.84	1.69	1.87	1.85	1.72	1.88
	3	1.86	1.73	1.89	1.93	1.84	1.89	1.98	1.92	1.90
	2	2.00	1.95	1.90	1.82	1.65	1.99	1.68	1.45	2.05
	1	1.64	1.35	2.07	-	-	-	-	-	-

Figure 5.8: Comparison of the influence of the skewness correction on the convergence of velocity and pressure.

The influence of the local grid irregularities on the error of the velocities is very significant without the correction. The number and strength of the error streaks increase considerably without the correction (see Figs. B.10, B.11, B.30 and B.31). Similarly, the local error for pressure without the correction shows particularly high levels around the grid irregularities (see Fig. B.32). The correction reduces the errors and the sizes of these local features (see Fig. B.12). A characteristic feature of these localised errors is the spread of the error into sectors. Without the correction the localised errors reduce with a less-than-theoretical order, also reducing the observed order based on the global error norm.

5.4.6 The influence of the boundary condition discretisation



	i	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
Gradient based	5	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
	4	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
	3	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
	2	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
	1	2.51	2.24	2.56	-	-	-	-	-	-
Node based	5	2.03	2.03	2.00	2.05	2.00	2.02	2.04	2.01	2.01
	4	2.05	2.00	2.03	2.10	1.98	2.06	2.08	1.99	2.05
	3	2.11	1.98	2.07	2.25	2.02	2.15	2.18	2.00	2.11
	2	2.26	2.02	2.15	2.38	2.21	2.30	2.33	2.12	2.23
	1	2.39	2.22	2.32	-	-	-	-	-	-

Figure 5.9: Comparison of the influence of the discretisation of extrapolation-type boundary conditions on the convergence of velocity and pressure.

The comparison of the grid convergence for the two discretisations of the extrapolation-

type boundary conditions is shown in Fig. 5.9. The differences between the discretisations are minor. However, the observed order of accuracies for the opposite node-based extrapolation seems to be closer in general to the theoretical order of the method than with the gradient-based extrapolation. This is particularly obvious for the pressure. On the other hand, the error norms in the case of the node-based extrapolation are consistently larger, with the largest difference also being observed for the pressure.

The distributions of local error largely confirm these findings (see Figs. B.10-B.12 and Figs. B.34-B.36). The scaled distributions are very similar and the differences are generally minor. Because inflow- and outflow-type boundary conditions are used, the differences in the discretisation are limited to the top and right-hand boundaries for the velocity components and to the bottom and left-hand boundaries for the pressure. This is also visible in the local error distributions. For the velocity components the clearest differences can be observed close to the right-hand and top boundaries, particularly for the region of large error close to the top right-hand corner (see Figs. B.10, B.11, B.34 and B.35). For the pressure the increase in the error in the middle of the domain clearly extends further towards the bottom right-hand corner in the case of the node-based extrapolation (see Figs. B.12 and B.36). Another striking feature of the pressure is the high level of error in the row of cells at the bottom boundary, compared to the rest of the domain, for the gradient-based extrapolation. A similar increase in error is also observed in the case of the opposite node-based extrapolation, but the increase is significantly more modest.

6 The verification of the time accurate free surface solver

In the second part of the code verification exercise the grid and time step convergence of the free surface solution were studied using an unsteady free surface flow based on potential flow theory. As the bulk flow implementation has been verified separately, this study concentrates on the verification of the spatial and temporal discretisations of the free surface boundary condition. Additionally, the behaviour of the numerical error and, particularly, the behaviour of the error in the Fourier components of the free surface wave are studied in detail.

6.1 The manufactured solution and the source terms

As in the first verification exercise, the governing field equations for the flow are the incompressible Euler and continuity equations. However, because of the presence of the free surface, the kinematic and dynamic boundary conditions also have to be considered in the evaluation of the manufactured source terms. The solution of the free surface flows is based on the arbitrary Lagrangian-Eulerian form of the Euler equations (2.14). However, the manufactured source terms can be evaluated in the Eulerian frame of reference using form (2.7). The source terms are then substituted into the ALE form of the equations (2.14) using Eq. (4.5). Just as in the first exercise, the differential forms of the field equations are used. The set of equations is thus

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + u\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} \tag{6.1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + v\frac{\partial u}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} \tag{6.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{6.3}$$

$$\left[1 + \left(\frac{\partial \eta}{\partial x}\right)^2\right]^{-1/2} \left(\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} - v\right) = 0 \tag{6.4}$$

$$p|_{fs} - \rho g \eta = 0 \tag{6.5}$$

for the Euler equations, the continuity condition, the kinematic boundary condition, and the dynamic boundary condition, respectively. Here $p|_{fs}$ is the piezometric pressure on the

unknown free surface $\eta(x,t)$. As the gravity is taken into account with the piezometric pressure, it does not appear explicitly in the Euler equations. Furthermore, because the free surface location is defined in this case as a function of x, the kinematic boundary condition (2.18) has to be written in the form (6.4). Here the multiplier in front is the length of the normal vector, and the term in the parentheses consists of the difference between the dot products of the flow velocity and the surface velocity with the components of the surface normal $(-\partial\eta/\partial x, 1)$.

The manufactured solution used in this study is for a standing wave in a rectangular container based on the linearised potential flow theory. There are several reasons for the choice of this particular solution. First, it is given by smooth analytical functions activating not only every term in the governing equations but also every derivative in the error expansion. As such, the solution is particularly suitable for MMS. Second, the solution has physical realism. This makes it easier to interpret the influence of numerical error in terms of practical physical changes in the solution. Other advantages offered by the choice of manufactured solutions possessing physical realism are mentioned by Eça et al. (2006, 2007). First, the terms are exercised in a manner similar to a real problem, highlighting similar difficulties. Second, the source terms are smaller, thus avoiding degenerate forms of altered equations. Third, it makes the methodology more attractive for the engineering community.

The flow field (u,v,p) and the free surface shape η for the standing wave are given by (Paterson, 1983)

$$u_{ms}(x,y,t) = \frac{\eta_0\omega}{\sinh(kh)}\cosh[k(y+h)]\sin(kx)\sin(\omega t) \tag{6.6}$$

$$v_{ms}(x,y,t) = -\frac{\eta_0\omega}{\sinh(kh)}\sinh[k(y+h)]\cos(kx)\sin(\omega t) \tag{6.7}$$

$$p_{ms}(x,y,t) = \frac{\rho g\eta_0}{\cosh(kh)}\cosh[k(y+h)]\cos(kx)\cos(\omega t) \tag{6.8}$$

$$\eta_{ms}(x,t) = \eta_0\cos(kx)\cos(\omega t) \tag{6.9}$$

Here $k = m\pi/L$, $\omega^2 = gk\tanh(kh)$, η_0 is the specified amplitude of the standing wave, h is the depth of the container, L is the length of the container, and m is an integer constant.

The manufactured source terms are obtained by substituting the solution (6.6)-(6.9) into

the governing equations (6.1)-(6.5). This gives

$$q_u = \frac{\eta_0^2 g k^2}{\sinh(2kh)} \sin^2(\omega t) \sin(2kx) \tag{6.10}$$

$$q_v = \frac{\eta_0^2 g k^2}{\sinh(2kh)} \sin^2(\omega t) \sinh[2k(y+h)] \tag{6.11}$$

$$q_m = 0 \tag{6.12}$$

$$Q_{kin} = \frac{-\eta_0 \omega \sin(\omega t)}{\sqrt{1 + [\eta_0 k \sin(kx)\cos(\omega t)]^2}} \left(\cos(kx) \left\{ 1 - \frac{\sinh\left[k(h+\eta_{ms})\right]}{\sinh(kh)} \right\} + \eta_0 k \sin^2(kx) \cos(\omega t) \frac{\cosh\left[k(h+\eta_{ms})\right]}{\sinh(kh)} \right) \tag{6.13}$$

$$Q_{dyn} = \rho g \eta_{ms} \left[\cosh(k\eta_{ms}) + \sinh(k\eta_{ms}) \tanh(kh) - 1 \right] \tag{6.14}$$

Here η_{ms} is the wave height (6.9) from the manufactured solution. The source term q_m for the continuity condition vanishes as the potential flow solution itself is based on the satisfaction of the condition. The source terms for the momentum equations, on the other hand, are produced purely by the advection terms as the manufactured pressure is such that the pressure gradient cancels the unsteady terms $\partial v_i / \partial t$. The source terms for the free surface boundary conditions are non-zero, as the manufactured solution is based on linearised boundary conditions to be satisfied at the still water level $y = 0$ rather than on the exact boundary conditions to be satisfied on the actual free surface.

The finite volume integrals of the distributed source terms for the field equations (q_u, q_v, q_m) are evaluated in the code using the discrete approximation (4.5). The source terms for the boundary conditions (Q_{kin}, Q_{dyn}) are substituted directly on the right-hand sides of the kinematic and dynamic boundary conditions (2.116) and (2.24), respectively.

6.2 The simulation case

The length and water depth of the container are $L = 40$ and $h = 1.6$, respectively. The free parameters of the manufactured solution have been set to $\eta_0 = 0.2$ and $m = 4$, giving two waves over the length of the tank. On the basis of the manufactured solution, a symmetry boundary condition was applied to the vertical walls of the tank, whereas a slip boundary condition was applied to the bottom boundary (see Sec. 2.7). For the slip condition the Taylor extrapolation-based approach (2.179) was used. With these conditions, the only explicit information taken from the manufactured solution was that the flow is

parallel to the boundaries, i.e. the normal component of velocity vanishes at these boundaries. Extrapolation is used for the other flow variables. As the flow is non-vortical, the condition (2.191) based on the assumption of zero vorticity was used on the free surface in the extrapolation of the velocity components into the ghost cells.

The choice of discretisations used in the study corresponds to the one on the first line of Tab. 5.2. In addition, a third-order upwind scheme is used for the discretisation of the free surface slope. This is equal to the choice $k = 1/3$ in Eqs. (2.62) and (2.63).

Only one set of grids was used in this exercise. Here, the refinement was performed by generating each grid separately with Delaundo. Thus, for each successive grid in the set the boundary point distributions were refined and the grid was generated with Delaundo using the same control parameters. The left half of the coarsest grid with maximum free surface deformation is shown in Fig. 6.1. For the grid updating the simple algebraic smoothing process was used.

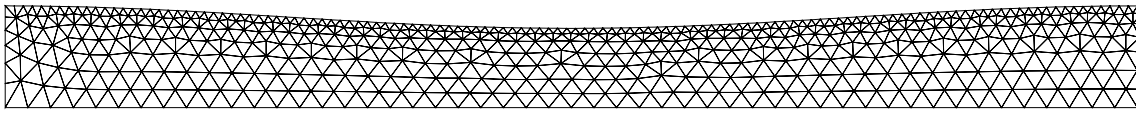


Figure 6.1: The left half of the coarsest grid at $t = 0$.

A refinement factor of roughly $r = \sqrt{2}$ was used for both the boundary nodes and the time step. The case was simulated with six grids and six time steps for one oscillation period. The number of faces on the free surface N_{fs} and the total number of elements N_e for the different grids are given in Tab. 6.1. The number of time steps per one period N_T is given in the same table.

Table 6.1: The number of free surface faces and the total number of elements, as well as the number of time steps per one oscillation period for different cell size and time step refinement levels.

Level	0	1	2	3	4	5
N_{fs}	250	353	500	706	1000	1413
N_e	1980	4243	8320	17040	33630	67845
N_T	50	71	100	142	200	284

6.3 The analysis methodology

Similarly to the steady verification case, the quantitative order of accuracy studies were based on the L_2 error norm. However, only the error in the wave height was considered. As in this case the simulation involves both spatial and temporal scales, the norm is taken as

$$err_2(\eta) = \frac{\sqrt{\sum_{l=1}^{N_{fs}} \sum_{k=1}^{N_T} \left(\eta_l^k - \eta_{ms}(x_l^k, t^k)\right)^2}}{\sqrt{N_{fs} N_T}} \,. \tag{6.15}$$

Here η_l^k and $\eta_{ms}(x_l^k, t^k)$ are the numerical and manufactured y-components of the wave height, respectively, at the centre of the free surface face l on the kth time step.

The order of accuracy was determined using the same procedure as in the steady case. Thus, two types of fits were used to determine the free parameters of the one-term estimate (5.11) for the error. In the fits, results on two or three levels were used. With unsteady cases the manufactured solution can only be reached if both the cell size and the time step approach zero at the same time. The order of accuracy studies were, therefore, based on a set of cases in which both discretisation parameters were refined simultaneously with (nearly) the same ratio ($r = \sqrt{2}$). The inverse of the number of free surface faces per wave length $2/N_{fs}$ and the inverse of the number of time steps per oscillation period $1/N_T$ were used as the discretisation measures.

The drawback with a global error norm is that it does not reveal anything about the spatial or temporal variation of the local error. Therefore, the global order of accuracy study was supplemented with a more detailed analysis of the behaviour of the numerical error. However, the comparison of the local error with different discretisation parameters is complicated by the fact that the numerical error consists of both amplitude and phase errors (see e.g. Burg and Erwin, 2009). For particular locations and instants in time, this leads to non-monotonous convergence of the numerical error.

In order to circumvent these problems, several authors, e.g. Hoekstra et al. (2000) and Carrica et al. (2006), suggest an alternative approach based on the convergence study of the Fourier components of the numerical solution. In this work, the Fourier analysis was performed for the spatial instantaneous wave profiles, resulting in time histories of the Fourier coefficients. The Fourier coefficients for a component wave with a wave number

jk were evaluated with a discrete approximation of the continuous Fourier integrals

$$a_j(t) = \frac{2}{L} \sum_{l=1}^{N_{fs}} \int_l I_2[\eta(t) \cos(jkx)]\, dx \tag{6.16}$$

$$b_j(t) = \frac{2}{L} \sum_{l=1}^{N_{fs}} \int_l I_2[\eta(t) \sin(jkx)]\, dx \ . \tag{6.17}$$

Here k is the wave number of the first harmonic wave component and j is an integer constant. The integral is taken successively over each free surface face l. The wave height over each face is approximated using a second-order interpolation operator, I_2. The operator is based on the values at the centres of the face l and the two neighbouring faces. At the ends of the domain the values at two neighbouring faces on the same side of face l were used.

The manufactured solution has only one spatial wave component

$$\eta(x,t) = \eta(t) \cos(kx) = a_{1,ms}(t) \cos(kx) \ . \tag{6.18}$$

This implies that it would be sufficient to study the behaviour of just this component. However, further analysis shows that higher-order wave components also emerge into the numerical solution. Therefore, the first three Fourier components ($j \in \{1,2,3\}$) were considered, with the wave number of the first harmonic component based on the manufactured wave. Because of the form of the manufactured solution and as confirmed by the analysis, the Fourier coefficients b_j corresponding to the sine function are negligible and, thus, they have been discarded from the following discussion.

The first Fourier component was studied more carefully, whereas the study of the other two components was of a more qualitative nature. On the basis of the manufactured solution (6.9), the time variation of the first Fourier component is given by

$$a_{1,ms}(t) = \eta_0 \cos(\omega t) \ . \tag{6.19}$$

For the quantitative analysis of the first Fourier component of the numerical solution the temporal variation is approximated using a harmonic function with exponentially varying amplitude and with a period not necessarily equal to the period of the manufactured wave

$$a_1(t) \simeq \eta_0 e^{-\alpha t} \cos[(\omega - \beta)t] \ . \tag{6.20}$$

The choice of this functional form was based on the influence of numerical dispersion and damping on a propagating wave. A standing wave can be represented as two waves

of equal amplitude and wave number which travel in opposite directions. The amplitude factor α and the phase factor β were solved by a non-linear fit of the function (6.20) to the first Fourier coefficient time histories.

For $\alpha t, \beta t \ll 1$ the time variation of the numerical error in the first harmonic component can be approximated further. By using Taylor series for the numerical solution defined with Eq. (6.20) the error $a_1(t) - a_{1,ms}(t)$ can be approximated with

$$\begin{aligned}
&\eta_0 e^{-\alpha t}\cos[(\omega-\beta)t] - \eta_0\cos(\omega t) \\
&= \eta_0 e^{-\alpha t}\left[\cos(\omega t)\cos(\beta t) + \sin(\omega t)\sin(\beta t)\right] - \eta_0\cos(\omega t) \\
&= \eta_0\left[1-\alpha t + O((\alpha t)^2)\right]\left\{\cos(\omega t)\left[1-O((\beta t)^2)\right]\right. \\
&+ \left.\sin(\omega t)\left[\beta t - O((\beta t)^3)\right]\right\} - \eta_0\cos(\omega t) \\
&= \eta_0\beta t\sin(\omega t) - \eta_0\alpha t\cos(\omega t) \\
&- O((\alpha t)(\beta t)) - O((\beta t)^2) + O((\alpha t)^2)
\end{aligned} \tag{6.21}$$

For small values of αt and βt the first-order terms dominate. In this case a simplified approximation for the error can be written with just one trigonometric function as

$$\begin{aligned}
a_1(t) - a_{1,ms}(t) &\simeq A(t)\sin(\omega t - \theta)\ \text{, where} \\
A(t) &= \eta_0 t\sqrt{\alpha^2+\beta^2} \\
\theta &= \tan^{-1}\frac{\alpha}{\beta}\ .
\end{aligned} \tag{6.22}$$

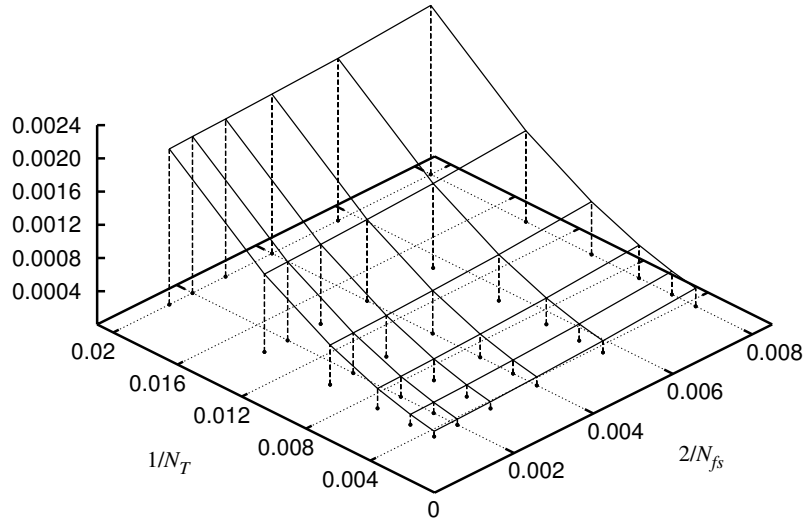
The fitting of the exact approximation (6.20) to the numerical solution data was performed with $\omega - \beta$ instead of β as the free parameter. However, for clarity of illustration the results are generally presented for β rather than for the actual oscillation frequency. The reason for this is that in this case both α and β should approach zero with discretisation refinement.

6.4 The results

The main results for the time-accurate free surface case are presented and discussed in this section. A full set of results is included in the Appendix (see Appendix C).

6.4.1 Global error norm

The scaled L_2-norms of the error in wave height for all of the simulated cases are presented in Fig. 6.2. The results show monotonous convergence for every grid level with



		N_{fs}					
		1413	1000	706	500	353	250
N_T	284	**6.44E-05**	7.03E-05	8.13E-05	1.03E-04	1.41E-04	2.18E-04
	200	1.23E-04	**1.30E-04**	1.41E-04	1.63E-04	2.01E-04	2.80E-04
	142	2.39E-04	2.46E-04	**2.57E-04**	2.79E-04	3.17E-04	3.97E-04
	100	4.76E-04	4.84E-04	4.95E-04	**5.16E-04**	5.54E-04	6.35E-04
	71	9.37E-04	9.46E-04	9.57E-04	9.78E-04	**1.01E-03**	1.10E-03
	50	1.87E-03	1.88E-03	1.90E-03	1.92E-03	1.95E-03	**2.03E-03**

Figure 6.2: The L_2-norm of the error in wave height $err_2(\eta)$ with different grid and time step combinations. The set of cases chosen for the order of accuracy study is marked in bold.

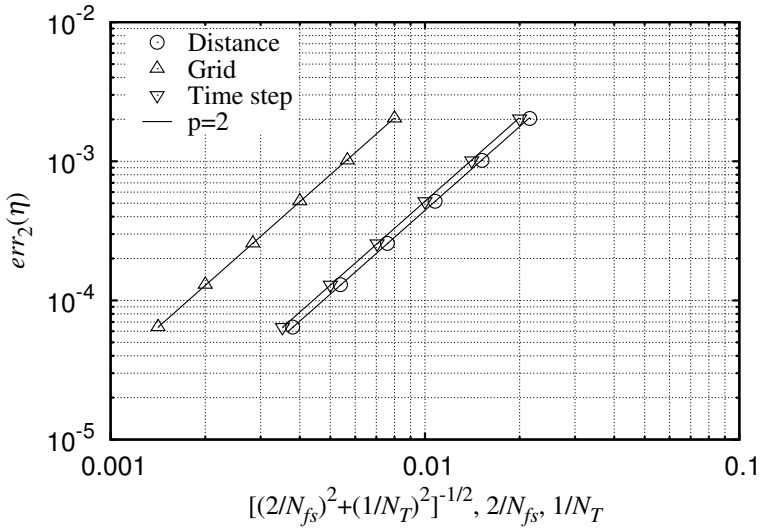
time step refinement (columns) and for every time step with grid refinement (rows). It can also be seen that within the tested parameter space the changes between successive levels with the time step refinement are significantly larger than with the grid refinement. The observed behaviour can be explained by the differences in the spatial and temporal discretisation resolutions, measured as faces per wave length or as time steps per oscillation period. For example, with the finest grid and the smallest time step the spatial resolution

is over twice as high as the temporal resolution. Thus, it can be assumed that within the tested range the temporal discretisation error is larger than the spatial discretisation error and, therefore, has a greater influence on the total error norm.

The results in Fig. 6.2 confirm that both discretisation parameters have to be refined at the same time in order to reach the manufactured solution (see Sec. 6.3). If only one of the parameters is refined, the discretisation error associated with the unrefined parameter will remain. This was also briefly studied quantitatively by performing a non-linear least-squares fit using all of the cases with the error approximated by

$$f(\Delta\tilde{x}, \Delta\tilde{t}) = C_x \Delta\tilde{x}^{p_x} + C_t \Delta\tilde{t}^{p_t} \tag{6.23}$$

The fit for the results in Fig. 6.2 gives $p_x = 1.88$, $C_x = 1.43$, $p_t = 2.00$, $C_t = 4.60$. This suggests that the method is second-order accurate in both space and time.



	Distance			Grid			Time Step		
i, j	$p_\eta^{i,2}$	$p_{\log(\eta)}^{i,3}$	$p_\eta^{i,3}$	$p_\eta^{i,2}$	$p_{\log(\eta)}^{i,3}$	$p_\eta^{i,3}$	$p_\eta^{i,2}$	$p_{\log(\eta)}^{i,3}$	$p_\eta^{i,3}$
5	2.00	2.00	1.99	2.03	1.99	1.98	2.00	2.00	2.00
4	1.99	1.99	1.99	1.96	1.99	2.01	2.00	1.99	1.99
3	1.99	1.98	1.98	2.02	1.98	1.96	1.99	1.98	1.98
2	1.97	1.98	1.98	1.94	1.98	1.99	1.97	1.98	1.98
1	1.99	-	-	2.01	-	-	1.98	-	-

Figure 6.3: The L_2-norm of the error in wave height with simultaneous grid and time step refinement and the observed orders of accuracies using results on two ($p_\eta^{i,2}$) or three ($p_{\log(\eta)}^{i,3}$ and $p_\eta^{i,3}$) successive levels.

However, a more thorough study of the order of accuracy of the method was also performed on the basis of a set of results with simultaneous grid and time step refinement (the bold entries of the table in Fig. 6.2). As both parameters are refined with (almost) the same ratio and as the assumed order of accuracy relative to both parameters is the same, the study can be based on either parameter or a combination of these two. Fig. 6.3 shows the convergence of the numerical error as a function of three different parameters and the corresponding observed orders of accuracies. In addition to the discretisation parameters based directly on the number of free surface faces and time steps, results are also presented for a fit using a parameter based on the distance from the origin of the parameter space. Just as in the steady case, three kinds of fits were performed for each parameter, one using results on two levels and two using results on three levels.

In every case the observed order of accuracy is very close to the assumed theoretical order of accuracy $p = 2$ and thus the free surface solution method is verified as being second-order accurate. If the results are studied in more detail, slightly greater deviations from the theoretical order are observed for the fits based on the number of free surface faces. Furthermore, in this case the fit with two levels results in an oscillatory order of accuracy, whereas in the other cases monotonous convergence towards the theoretical order is mainly observed.

The oscillations are related to two factors. First, the temporal discretisation error has a greater influence on the total error within the tested parameter space. Thus, the variation in the total error is more accurately measured using the parameter based on the number of time steps. Second, as shown in Tab. 6.2 the refinement ratios for successive refinement levels oscillate around $r = \sqrt{2}$. The oscillations are out of phase and, thus, the parameter based on the number of free surface faces is either too small or too large compared to the primary source of the discretisation error.

Table 6.2: The refinement ratios for the grid pairs with simultaneous grid and time step refinement.

Pair	1–0	2–1	3–2	4–3	5–4
r_x	1.412	1.416	1.412	1.416	1.413
r_t	1.420	1.408	1.420	1.408	1.420

6.4.2 Local error

The behaviour of the local error with grid refinement confirms the global error norm-based conclusion that the method is second-order accurate (see Appendix C). Here and hereafter the local error refers to the error of $\eta(x,t)$ in the (x,t)-space. By studying the local error with different levels of simultaneous grid and time step refinement it can be seen that the primary features of the local error scale to the second power of the refinement. As an

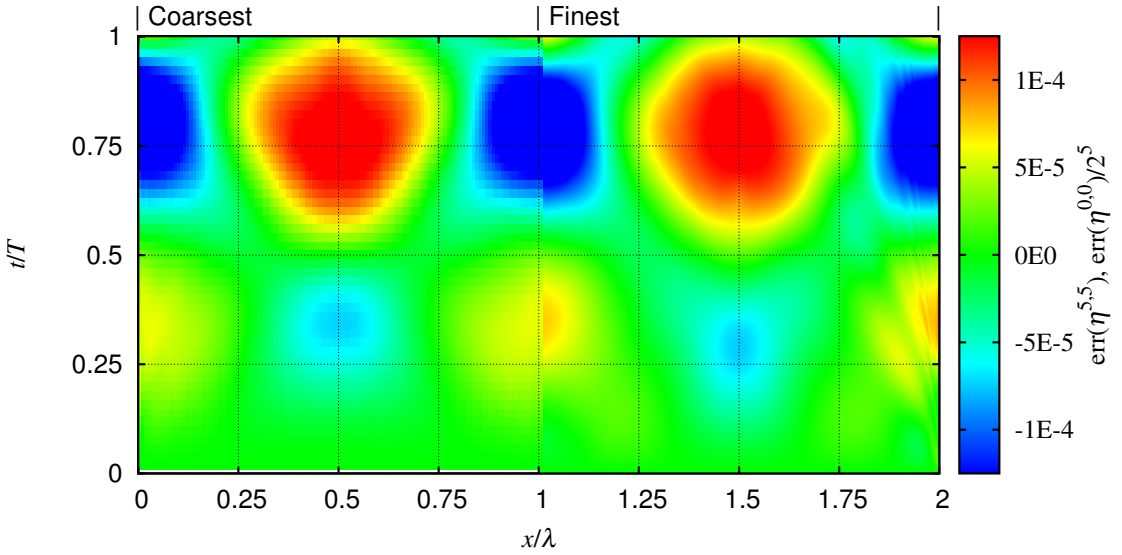


Figure 6.4: Local error with the coarsest and finest discretisation parameters. These are denoted by $err(\eta^{0,0})$ and $err(\eta^{5,5})$, respectively. The coarse results have been scaled down to the colour range assuming second-order accuracy.

example Fig. 6.4 compares the spatial and temporal variation of the error in wave height for the coarsest and finest discretisations. The coarse results have been scaled to the same colour range assuming second-order accuracy. The results for all of the simulated cases are collected in Appendix C. The overall similarity of the local error distributions is obvious. Likewise, the results show clearly that the error scales very closely to the second power of the refinement. However, there is a phase difference between the results with different refinements as the maxima and minima of the error occur at slightly different times. Furthermore, the intensity and shape of the areas with large errors are slightly different.

With the finest refinement there are visible streaks of large errors close to the boundary of the domain. The streaks are caused by a disturbance created at the boundary. The error caused by the disturbance propagates into the domain, with different frequency compo-

nents of the disturbance travelling at different speeds. A more detailed study reveals that a similar disturbance is created in all of the cases. However, on coarser levels the overall level of local error increases and the error caused by the disturbance is, thus, harder to detect. The actual reason for the disturbance is unclear, but it indicates that the implementation of the boundary treatment should be studied further. For practical simulations the influence is, however, negligible.

The shape of the error distribution suggests that, in addition to an error related to the only Fourier component of the manufactured solution, additional wave components emerge into the numerical solution. An error having a single wave component would result in a regular error pattern with symmetrical features, whereas the observed error distributions feature somewhat non-symmetrical patterns.

The additional wave components can be explained by the interaction of the numerical solution and the manufactured source terms. The manufactured solution is the solution of a linearised problem and as such, does not satisfy the unaltered, fully non-linear, governing equations of the flow solver. The purpose of the manufactured source terms here is to suppress the influence of the non-linearities of the original equations. It is well known that the solution of the fully non-linear free surface problem includes higher-order wave components propagating with the velocity of the primary wave. Thus, in physical terms the purpose of the source terms is to suppress the higher-order components from the numerical solution.

This can also be shown with Fourier analysis of the source terms. Fig. 6.5 shows the temporal variation of some of the most significant Fourier components of the source term for the kinematic boundary condition (6.13). This confirms that the source term consists of several different Fourier components. The figure also shows that of these the most significant second-order $\cos(2kx)$ component is two orders of magnitude larger than any of the other components.

An analytical approximation for the variation of these Fourier components was also constructed by performing a Taylor series expansion for the source term (6.13) with respect to $k\eta_0$. This gives the second-order accurate approximation

$$Q_{kin} = \cos(kx)\tfrac{1}{16}\,\omega k^2\eta_0^3 \sin(2\omega t)\cos(\omega t) \tag{6.24}$$

$$+\cos(2kx)\tfrac{1}{2}\,\omega k\eta_0^2 \coth(kh)\sin(2\omega t) \tag{6.25}$$

$$+\cos(3kx)\tfrac{3}{16}\omega k^2\eta_0^3 \sin(2\omega t)\cos(\omega t) + O(k^3\eta_0^4)\ . \tag{6.26}$$

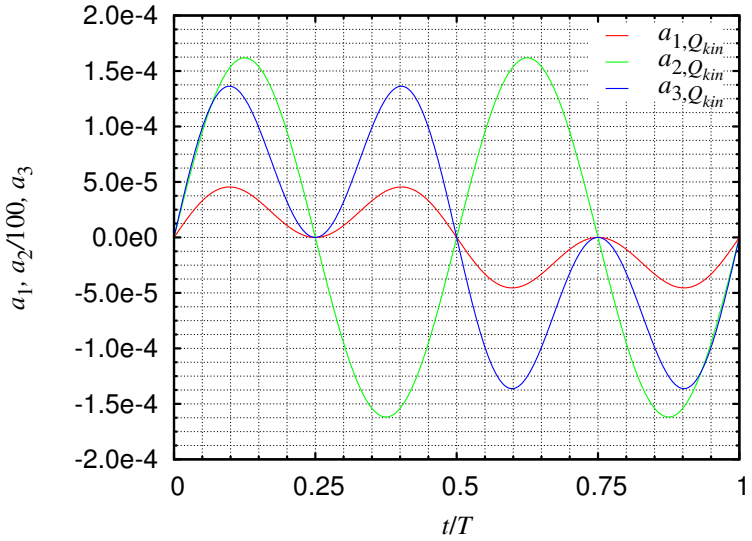


Figure 6.5: The temporal variation in the most significant Fourier components of the source term for the kinematic boundary condition (6.13).

On the basis of the approximation, the relative differences in the amplitudes of the different components can be compared easily for small values of $k\eta_0$. In the case at hand $k\eta_0 \simeq 0.063$.

In order for the higher-order components to be suppressed, the source terms and the numerical solution have to be in phase. However, because of the discretisation error the oscillation period of the numerical solution is slightly different from the manufactured one. Because of this the numerical solution and the source terms are increasingly out of phase and, thus, additional wave components emerge into the solution. This is easiest to realise by considering the instant at which the amplitude of the standing wave is at its maximum and the source term for the kinematic boundary condition should be zero. If the solution and the source term are out of phase, the source term is then non-zero and is generating additional wave components rather than suppressing them. The different wave components emerging in the process were studied by Fourier analysis of the instantaneous wave profiles.

6.4.3 The first-order wave component

The first Fourier component of the wave $\cos(kx)$ corresponds to the manufactured wave and is also the primary wave component of the numerical solution.

Qualitative verification

Fig. 6.6 shows the temporal variation in the error in the first component for separate grid refinement and time step refinement. In the cases presented the finest level is used for

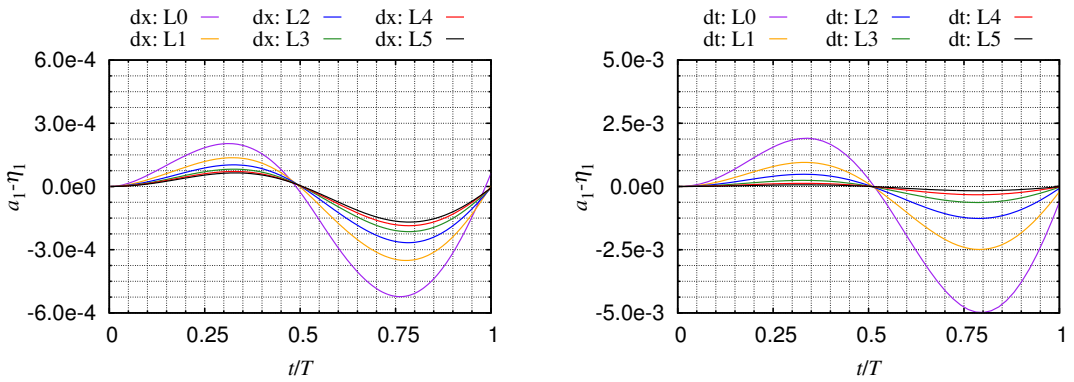


Figure 6.6: The temporal variation of the first Fourier component of the error with separate grid (left) and time step (right) refinement. For the fixed parameter the corresponding finest level has been used.

the fixed parameter. The figure shows the influence of the variation of just one discretisation parameter in the case of multi-variable discretisation. Monotonous convergence towards a specific solution is observed in both cases. However, the simulation result does not converge to the manufactured solution, but to a solution with a residual error. This is clearly visible with the grid refinement. In this case the residual error is caused by the temporal discretisation error. On the other hand, the case with time step refinement converges similarly to a solution with residual error caused by the spatial discretisation error. However, because the spatial discretisation error is considerably smaller than the temporal one within the tested range, this is not obvious from the graph.

The numerical solution converges towards the manufactured solution only if both discretisation parameters are refined at the same time. Fig. 6.7 shows the convergence of the first Fourier component of the error with simultaneous grid and time step refinement. The variation with refinement is very similar to the variation with just the time step refinement in Fig. 6.6. This is a further confirmation that, with the tested parameters, the error associated with the time step refinement is significantly greater than that associated with spatial discretisation. The results show convincingly that with simultaneous refinement the first Fourier component of the numerical solution converges towards the manufactured solution with the amplitude of the error halved with every refinement. This is expected

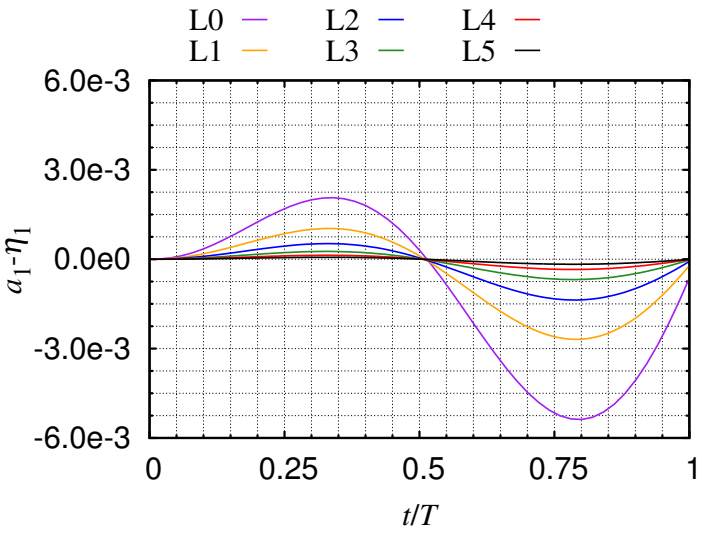


Figure 6.7: The temporal variation of the first Fourier component of the error with simultaneous grid and time step refinement.

behaviour with a second-order method.

However, as can be seen from the results in Figs. 6.6 and 6.7 the refinement has, in fact, two types of influence on the error. In addition to the influence on the amplitude of the error, the period of the oscillations of the error also changes with the refinement.

Quantitative verification

As explained in section 6.3, it has been assumed that the temporal variation of the first Fourier component of the error can be approximated using a harmonic function with exponentially varying amplitude (6.20) or with the simplified approximation (6.22). The two free parameters of the approximations – the amplitude factor α and the phase factor β – were determined by fitting the approximations to the first Fourier component time histories. On the basis of this, the influence of the refinements on the temporal variation of the error was also studied quantitatively.

Fig. 6.8 shows α and β for every combination of grid and time step refinement using the full approximation (6.20). The phase factor β shows monotonous convergence for both parameter refinements. The amplitude factor α, on the other hand, shows monotonous convergence only with the time step refinement. With the grid refinement oscillatory

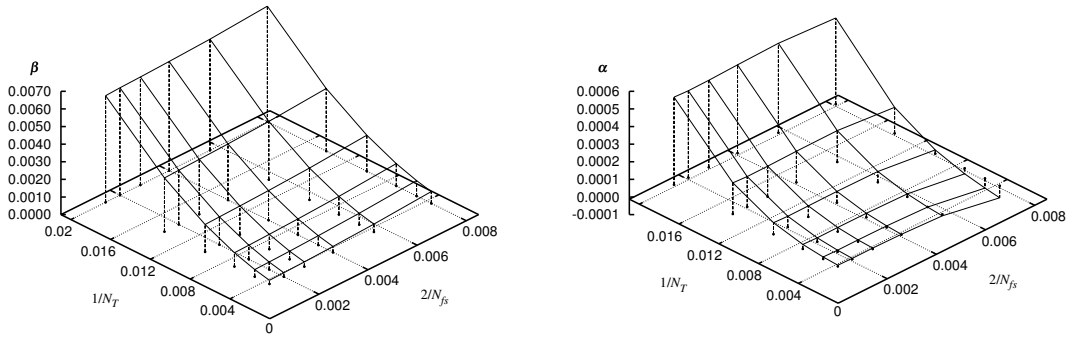


Figure 6.8: Phase factor β and amplitude factor α with different grid and time step refinements.

convergence is observed and the amplitude factors are negative for small time steps and coarse grids.

Normally one would expect positive amplitude factors α as a result of numerical damping resulting from even derivatives of the truncation error. Negative values would suggest that the method would be unstable, as the amplitude of the oscillation would increase exponentially (see Eq. 6.20). However, here the negative values and the oscillatory convergence are assumed to be the result of a complex interaction between the numerical damping, phase error, and the source terms. Going back to Fig. 6.5, it can be seen that the first Fourier component of the source term is also non-zero. As the numerical solution and the source terms are out of phase as a result of the phase error, the source term modifies the amplitude of the first Fourier component. In fact, if one takes a closer look at the convergence with grid refinement in Fig. 6.6, it can be observed that there is a noticeable increase in phase shift with decreasing grid resolution. This is particularly obvious for the coarsest grid. Thus, the interaction between the phase error and the manufactured source terms also has an influence on the amplitude of the solution. Because of this it is not straightforward to draw conclusions on the damping properties of the discretisations on the sole basis on the amplitude variation with refinement.

Nevertheless, the characteristic parameters α and β were studied in more detail by qualitatively assessing their orders of convergence. Similarly to the case of the L_2-norm of the error, the order of accuracy of convergence was studied using cases with simultaneous grid and time step refinement (the bold entries of the table in Fig. 6.2). Fig. 6.9 shows the convergence of β and α as functions of the three generalised discretisation parameters. The reference lines in the figures are based on the assumed orders of accuracy for the nu-

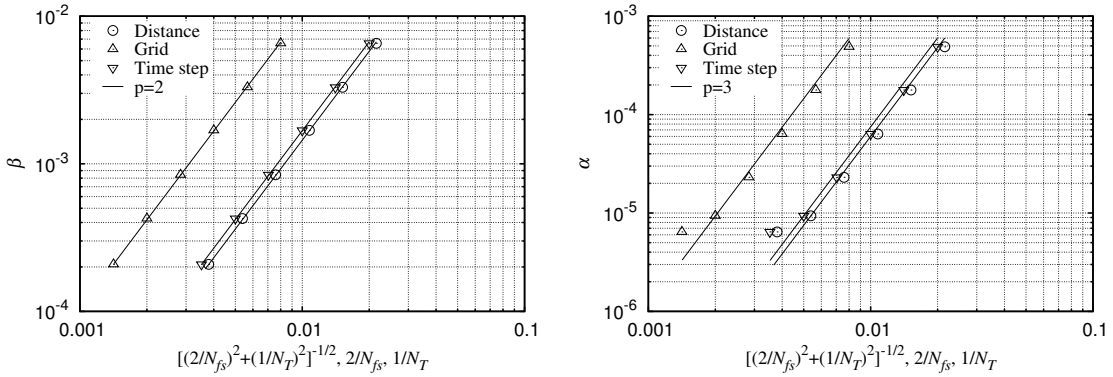


Figure 6.9: The convergence of the phase factor β and the amplitude factor α with simultaneous grid and time step refinement.

merical dispersion and numerical damping of the method, i.e. second-order for the phase and third-order for the amplitude.

The convergence of the phase factor is very accurately second-order. The amplitude factor, on the other hand, exhibits roughly third-order convergence for the coarser discretisations, but a significant drop in the order is observed for the two finest discretisations. The drop is believed to be related to the oscillatory convergence of the amplitude factor with the grid refinement.

Validity of the approximations of the temporal variation

The validity of the exponential-harmonic and the simplified approximations (6.20) and (6.22) has been checked in Fig. 6.10. The results are shown only for the simplified approximation as the two approximations are indistinguishable within the time range used. Four cases at the extremes of the refinement parameter space were considered by taking the coarsest and finest levels for both parameters and by combining these in the four possible ways.

The approximation performs extremely well. However, slightly greater deviation from the exact error is observed in the case of the coarsest grid combined with the finest time step. In the other cases the exact error is very accurately constrained within the sector defined by the approximate amplitude function $\pm A(t)$. Thus, the results in Fig. 6.10 suggest that the behaviour of the first Fourier component of the error can be described quite accurately with the two quantities α and β defined by either Eq. (6.20) or (6.22).

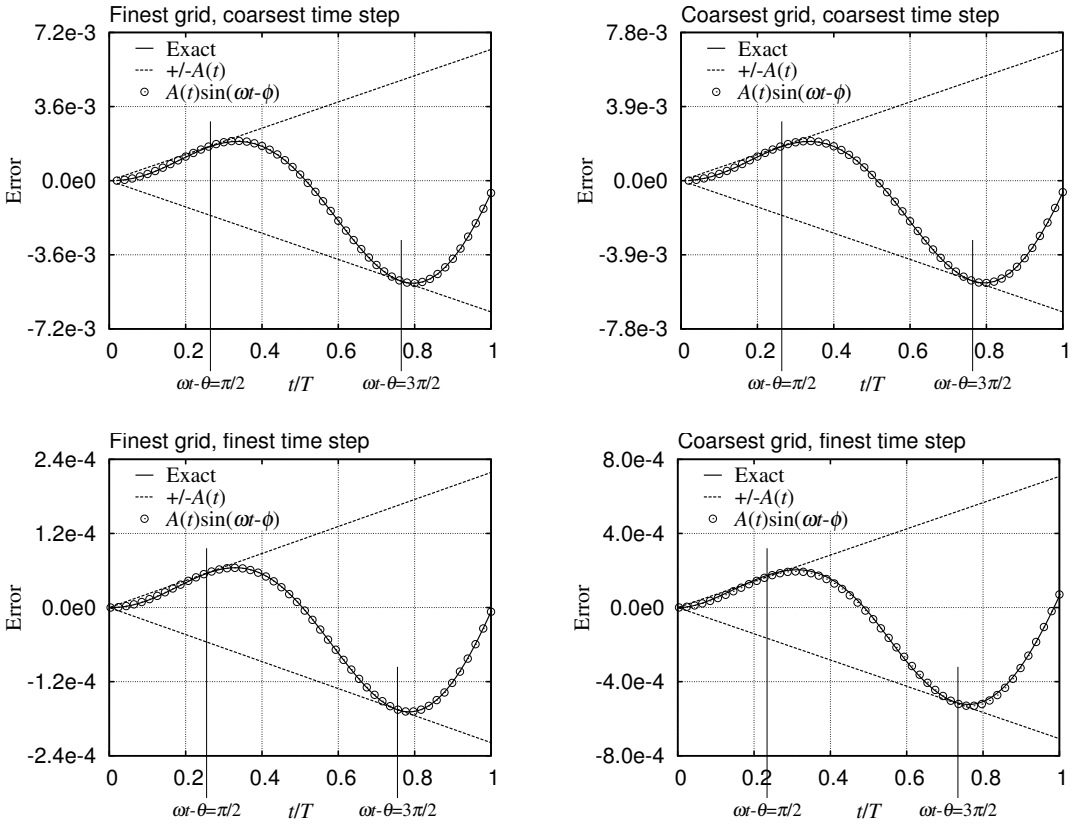


Figure 6.10: Comparison between the temporal variation in the first Fourier component of the exact error and the simplified approximation (6.22) with different grid and time step combinations.

The quantities α and β in the previous section were based on just one oscillation period ($T = 5.25$). In practical cases several oscillation periods are usually simulated. Thus, it is interesting also to study the validity of the approximation (6.22) in the case of a longer simulation. The validity was checked by repeating the simulations on the coarsest grid with the finest and the coarsest time steps over ten oscillation periods. These particular cases were chosen as the former showed the largest discrepancy between the exact error and the error approximation in Fig. 6.10 and as the latter corresponds to the largest discretisation error.

If the fit is performed over just one oscillation period, α is -6.8×10^{-5} and 4.9×10^{-4} and β is 6.7×10^{-4} and 6.6×10^{-3} with the finest and coarsest time steps, respectively. In the derivation of the simplified approximation (6.22) it was assumed that αt and βt are small values. With the aforementioned values the maximum values for the products over ten oscillation periods are $10\alpha T = -0.0036$ and 0.026 and $10\beta T = 0.035$ and 0.35 with the finest and coarsest time steps, respectively. Thus, the assumption of small values for the products is reasonable also in this case, particularly in the case of the finest time step.

Similarly to Fig. 6.10, the simplified approximation (6.22) is compared with the exact error in Fig. 6.11 over ten oscillation periods. The phase factor and the amplitude factor were evaluated by performing the fit over just the first oscillation period, as well as over ten oscillation periods. The approximations based on the first fit correspond to those presented in Fig. 6.10. As the results show, the simplified approximation also performs reasonably well over several oscillation periods. The fit based on just the first oscillation period increasingly overestimates the amplitude of the oscillation of the error. The oscillations are also increasingly out of phase. Nevertheless, in this case an error approximation based on just the oscillation amplitude is conservative and, thus, bounds the error over the whole range. With the fit over the full range, on the other hand, the simplified approximation is very close to the exact error. However, in this case larger deviations can be observed at the start of the simulation. Furthermore, the amplitude of the error oscillation is underestimated at certain locations.

There are three possible sources for the deviations between the approximated and the exact error observed above. The first one is the approximation of the temporal variation using the harmonic function with exponentially varying amplitude in Eq. (6.20). The second one is the simplification (6.22) based on the Taylor series expansion and neglect of the higher-order terms. The third one is the actual process of fitting the approximating function to the simulation data. In order to study the influence of different sources,

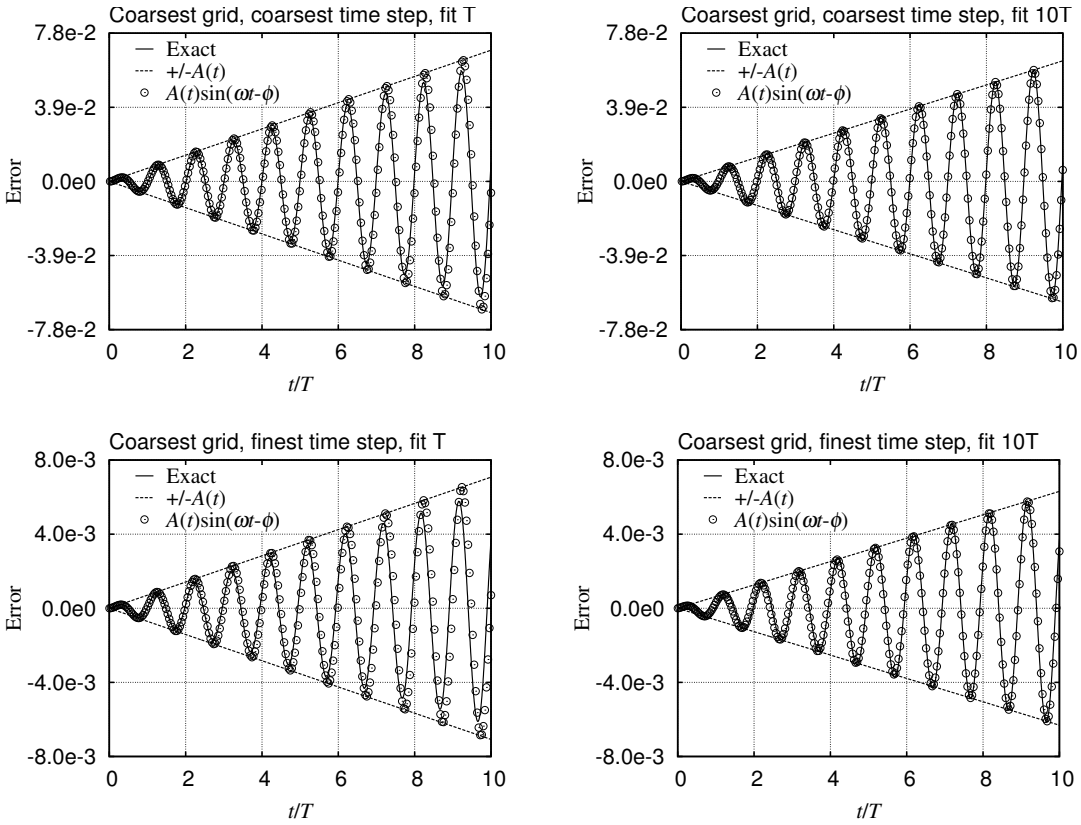


Figure 6.11: Comparison between the temporal variation in the first Fourier component of the exact error and in the error approximated with the simplified approximation (6.22) over ten oscillation periods. The fit was performed over the first period (left) and over ten periods (right).

the exponential-harmonic approximation (6.20) was fitted to the time history of the first Fourier component as well. Then the approximation errors of the exponential-harmonic and the simple approximations were compared. The results are shown in Fig. 6.12.

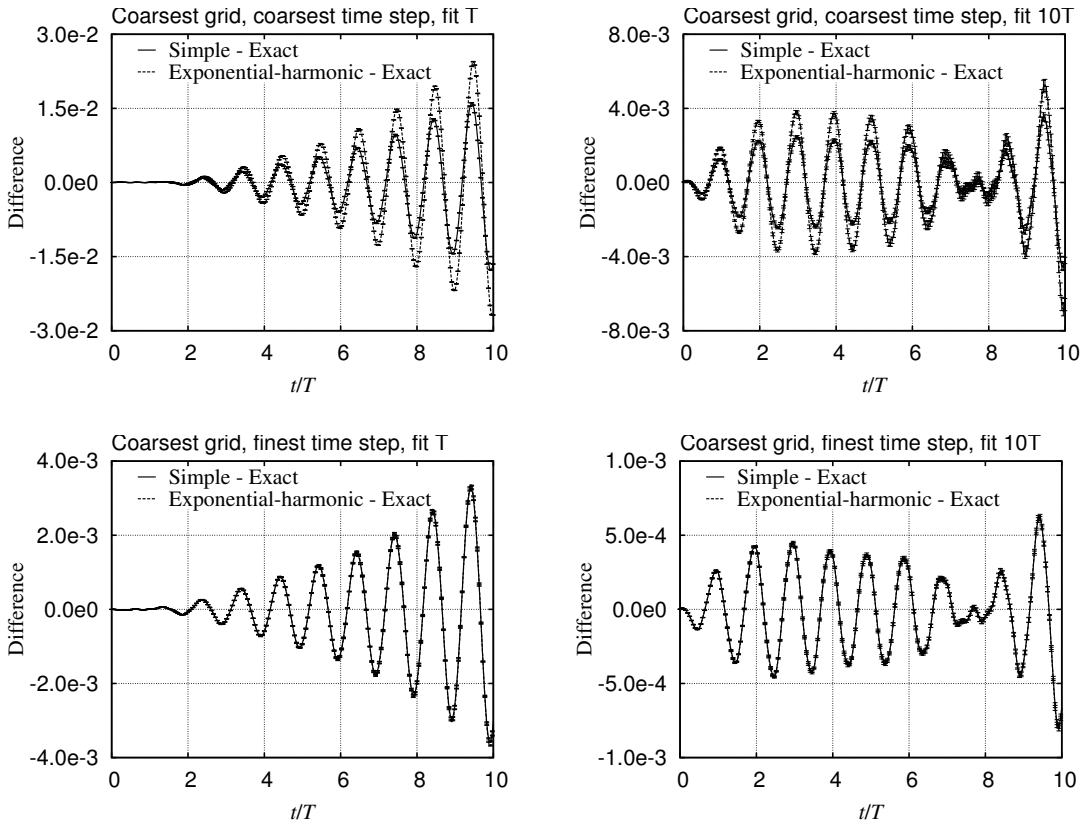


Figure 6.12: The comparison of the approximation error between the simplified error approximation (6.22) and the approximation based on the exponential-harmonic function (6.20). The uncertainty related to the actual fitting process is denoted by error bars. The cases are the same as in Fig. 6.11.

The possible deviations related to just the fitting process were approximated in Fig. 6.12 using the total differentials of Eqs. (6.20) and (6.22) with respect to changes in α and β. The uncertainties of the parameters α and β were estimated using the asymptotic standard errors of the parameters reported by gnuplot. Even if, in the case of a non-linear fit, these are not accurate estimates of the standard error (Gnuplot, 2007), it is believed that they give a correct indication of the order of magnitude of the uncertainties. On the basis of the results in Fig. 6.12, the order of magnitude of the fit-related uncertainties is negligible compared to the deviation between the approximations and the exact error. It is therefore assumed that the main contributions to the deviations come from the

exponential-harmonic approximation and from the truncated Taylor series expansion.

The primary source seems to be the approximation of the time variation of the numerical solution with the exponential harmonic function (6.20). Because the simplified approximation is based on the Taylor series expansion at $t = 0$ and the truncation error is a function of αt and βt, its influence only shows with large values of the parameters α and β. For the finest time step the parameters are small and thus in this case the approximations coincide. It is quite surprising that for the coarsest time step the simple approximation is closer to the exact solution than the exponential-harmonic approximation. However, this is probably a result of a favourable interaction of the truncation error and the approximation error in this particular case and should not be taken as a general conclusion for the superiority of the simple approximation.

In conclusion, the results in Figs. 6.10, 6.11 and 6.12 show that the time variation of the first Fourier component can be represented reasonably well but not exactly with the exponentially varying harmonic function (6.20) and the associated simplified approximation. One reason for the slight inaccuracy of the approximations is the harmonic forcing related to the manufactured source terms (see e.g. Fig. 6.5). This forcing always occurs at the prescribed frequency and amplitude related to the manufactured solution. As discussed above, the forcing is also one possible cause of the negative amplitude factors observed.

Estimation of the uncertainty

In addition to code verification, the approximations (6.20) and (6.22) can be used as a basis for error estimation in the standing wave case. A code verification, such as the one presented above, is performed on the basis of the assumption that the exact solution to the governing equations is known. However, typically the solution is not known and the error has to estimated. In order to demonstrate further the usefulness of the approximations for the temporal variation of the first Fourier component, the error was also estimated by using only the numerical solutions. For this the uncerainty estimation procedure presented by Eça and Hoekstra (2009) was used. This is partly based on the GCI (Grid Convergence Index) (Roache, 1998).

The error estimate for a particular combination of spatial and temporal discretisation was constructed by first estimating the errors related to the model parameters α and β with the least-squares GCI. The least-squares fits were performed using the available numer-

ical results with simultaneous grid and time step refinement (see Fig. 6.9). Because the observed order of both parameters was just below the theoretical orders, the Eça and Hoekstra (2009) procedure reduced to

$$U_{\phi,i} = 1.25\delta_{RE,i} + U_S \ , \tag{6.27}$$

where $U_{\phi,i}$ is the uncertainty of the numerical solution of a general variable ϕ_i on the discretisation level i and U_S is the standard error of the least-squares fit. The difference between the numerical solution and the Richardson extrapolated value ϕ_0, which corresponds to the infinite resolution, is

$$\delta_{RE,i} = \phi_i - \phi_0 = g_{\phi,p} h_i^p \ . \tag{6.28}$$

The estimate of the temporal variation in the error was evaluated on the basis of the total differential of the exponential-harmonic approximation (6.20) as

$$U_{a_1,i}(t) = \left| \frac{\partial a_1(t)}{\partial \alpha} \right| U_{\alpha,i} + \left| \frac{\partial a_1(t)}{\partial \beta} \right| U_{\beta,i} \ . \tag{6.29}$$

Here it was assumed that the parameters are fully correlated and the contributions were summed arithmetically. For uncorrelated parameters the contributions from the parameters would be summed in a least-squares sense (ISO, 1995).

If the exponential-harmonic approximation (6.20) is substituted into Eq. (6.29) and the same simplified approximations as in Eq. (6.21) are made, the uncertainty estimate becomes

$$U_{a_1,i}(t) = \eta_0 t |\cos(\omega_0 t)| U_{\alpha,i} + \eta_0 t |\sin(\omega_0 t)| U_{\beta,i} \ . \tag{6.30}$$

The comparison of this estimate and the actual error with the coarsest spatial and temporal discretisations over ten oscillation periods is shown in Fig. 6.13. As can be seen from the results, the uncertainty estimate is conservative and bounds the exact error over the whole range. The level of conservatism decreases as the time increases. This is related to the fact that the estimate is evaluated on the basis of the results over just the first oscillation period.

6.4.4 The second- and third-order wave components

As discussed before, additional higher-order wave components emerge into the numerical solution in addition to the primary wave component. Even if the amplitudes of these secondary components are much smaller than the amplitude of the primary component,

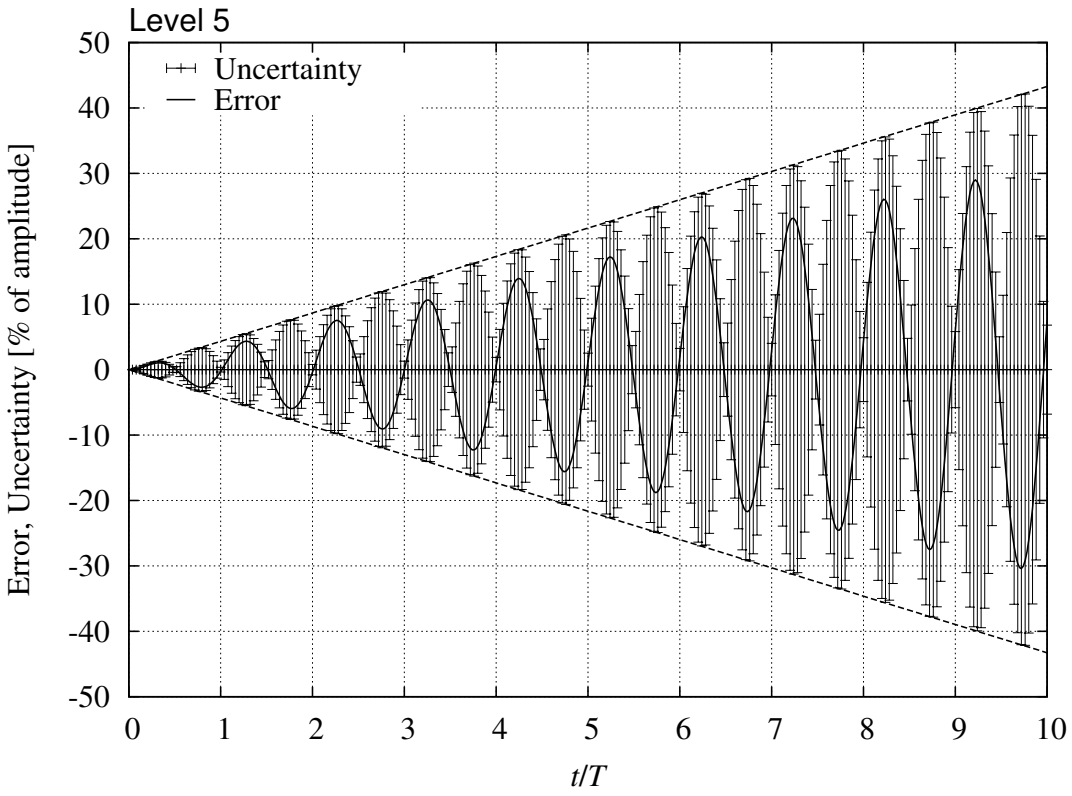


Figure 6.13: The comparison of the uncertainty estimate (6.30) and the actual error with the coarsest spatial and temporal discretisations. The dashed lines denote the limiting range of the uncertainty estimate, which is evaluated on the basis of the oscillation amplitude. For the sake of clarity of illustration only every second uncertainty bar is drawn.

the secondary components are still significant in terms of the error. Namely, one should bear in mind that, as the manufactured solution consists only of the primary component, the error related to the secondary components is equal to the components themselves. In the following discussion the first two higher-order components, i.e. the second and the third, are considered. The treatment is not as thorough as that of the primary component above, and the discussion is restricted to general observations on the behaviour of the higher-order components.

Fig. 6.14 shows the spatial and temporal variation in the local error when the first-order component has been removed. The results correspond to both the coarsest and the finest and the coarsest and the second-finest grid/time step pair. Similarly to the full local error in Fig. 6.4, the results for the coarser discretisations have been scaled to the range of the finest discretisation assuming second-order accuracy.

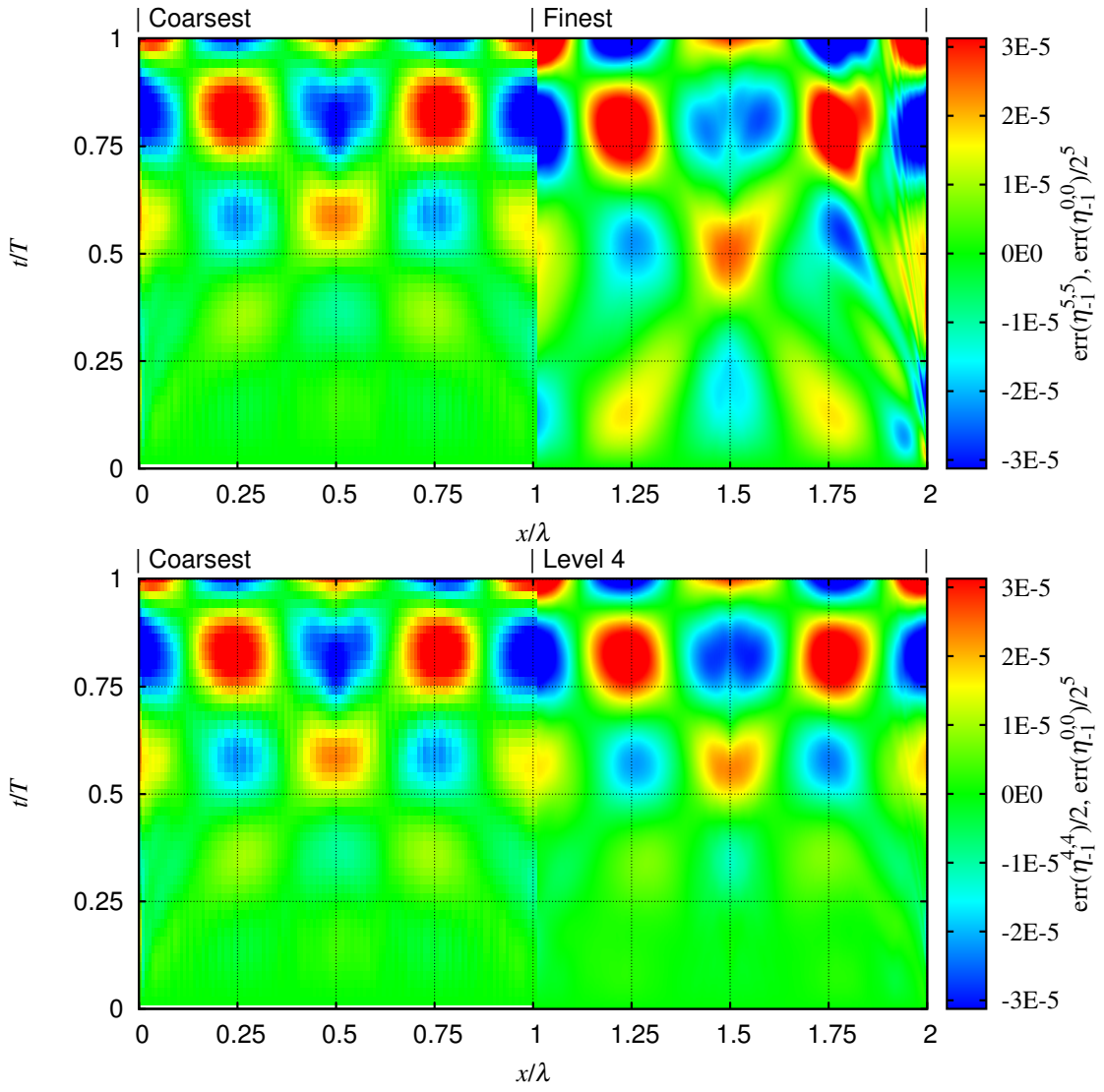


Figure 6.14: Local error with the first-order component removed. The results with the coarsest (Level 0) and finest (Level 5) discretisation parameters, as well as with the coarsest and the second finest (Level 4) ones. The results have been scaled down to the range of the finest discretisation assuming second-order accuracy.

The results show that in this case too the overall level of the local maxima and minima scale quite accurately to the second order of the refinement. However, while the scaled distributions for the coarsest and second-finest discretisations are very similar, the results for the finest discretisation show both a large phase shift and some additional features in comparison to the other distributions. The most obvious difference is the areas of large error around $t/T = 1/8$. Additionally, the streaks originating from the boundary are more clearly visible, as the first-order component has been removed from the error.

The behaviour of the higher-order components of the error was studied in more detail by examining the temporal variation in the Fourier components with different refinements. Fig. 6.15 shows the temporal variation of the second and the third Fourier components of the error with both separate and simultaneous grid and time step refinements.

The results for the second-order component are quite consistent, with similar convergence for all of the refinements. Particularly consistent convergence is observed with the time step refinement. The temporal variation in the second-order error is very similar to that for the first-order component. However, in this case the oscillation frequency is twice as high. Furthermore, the amplitude increases faster than linearly with time. The observed convergence of the third-order component with different refinements, on the other hand, is clearly inconsistent.

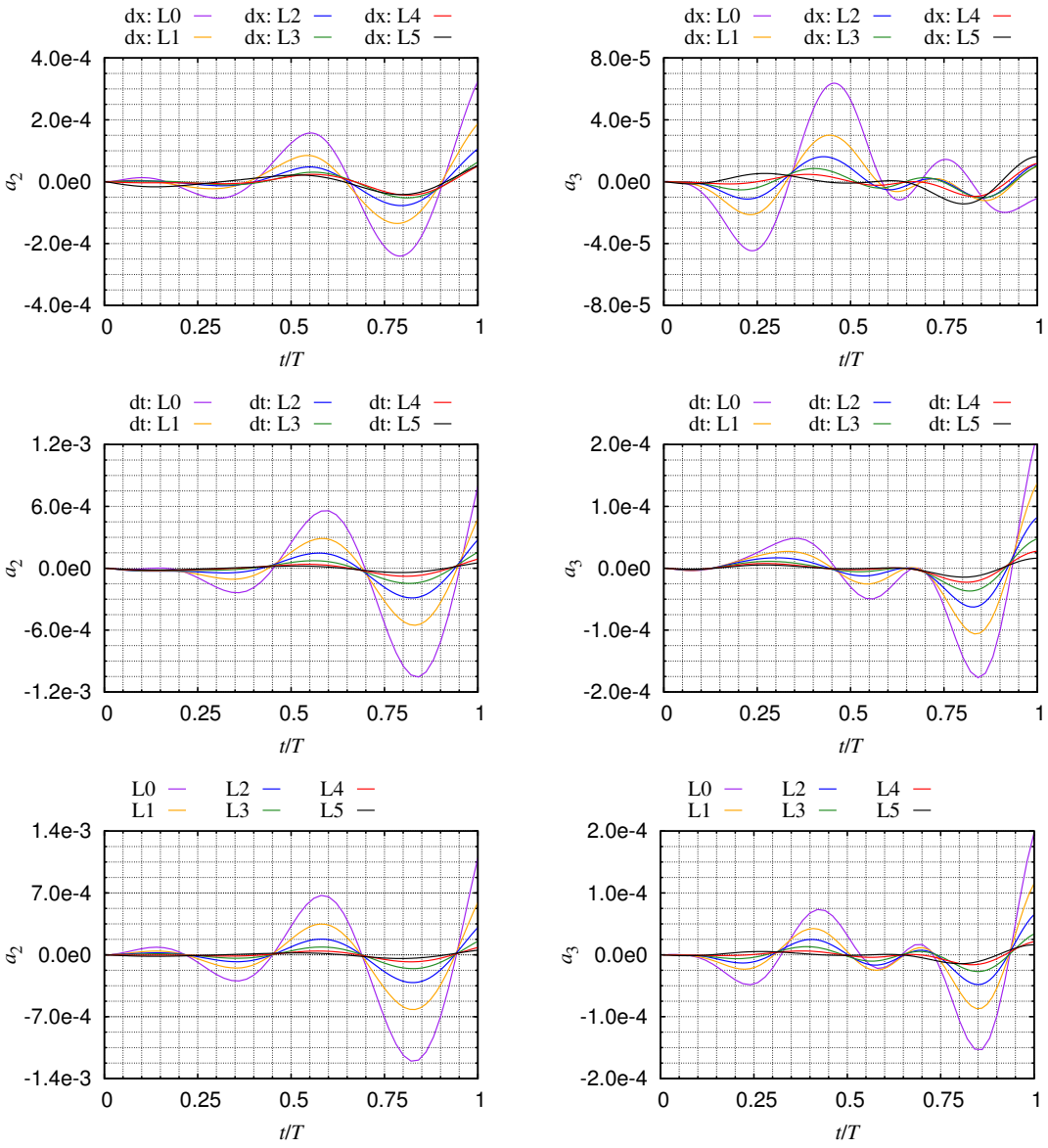


Figure 6.15: The temporal variation in the second (left) and the third (right) Fourier components with different refinements. With separate refinement the finest discretisation has been used for the fixed parameter. Top: grid refinement, middle: time step refinement, bottom: simultaneous grid and time step refinement.

7 Discussion on the code verification studies

The MMS has proven to be a powerful method, not only for showing consistency, but also for demonstrating the correct order of accuracy of a numerical method. Here, the method of manufactured solutions was used for the verification of a time-accurate, surface tracking, free surface flow solver. The study has shown that MMS can also be used easily and effectively for the verification of surface tracking free surface discretisations, i.e. with a highly non-linear boundary condition.

7.1 The verification of the flow solver

The bulk flow and time-accurate free surface discretisations were verified separately using two different manufactured solutions. In addition to the global error norm, the convergence of the local error was also considered. Consistency was shown in every case. Furthermore, in most cases the observed order approached the theoretical order of convergence with refinement, which verifies the corresponding implementations. However, the results of the first verification study indicate that there are issues with some options of the bulk flow solution method. The use of these options results in a less-than-theoretical order of convergence with extremely fine grids. The results show that with anisotropic grids the theoretical order of accuracy cannot be achieved without the skewness correction. Similarly, the direct evaluation of the pressure term in the momentum equations using a least-squares approximation of the pressure gradient results in growing deviation from the theoretical order of convergence with grid refinement.

During the study some unresolved problems with the free surface method also surfaced. For example, it was observed that the amplitude factor of the primary error component exhibits oscillatory convergence with grid refinement. The reasons for this should be studied further. Additionally, there seem to be some problems at the ends of the free surface boundary, where a localised disturbance develops and propagates into the domain.

With the grids used the generalised discretisation parameter $1/\sqrt{N_e}$ based on the number of elements in a grid was shown to be quite an accurate measure of the grid resolution, even if the grid sets used were based on two different refinement techniques. Furthermore, despite the fact that there are clear differences in the local error distributions, the refine-

ment technique did not have a significant influence on the observed order of accuracy. Orders very close to the theoretical orders were achieved with both techniques. Thus, it is assumed that both refinement techniques produce grid sets with sufficient geometric similarity. Local grid irregularities may, however, lead to slight differences.

7.2 Local behaviour of the discretisation error

In the case of unsteady free surface flow, in addition to the code verification based on a global error norm, a detailed study was performed to investigate the spatial and temporal variation in the error in wave height with different grid and time step refinements. The behaviour and generation of different error components were exposed by Fourier analysis of the instantaneous spatial waves. An approximate representation was constructed for the description of the temporal variation of the primary spatial error component. On the basis of the approximation it was demonstrated how it is also possible to study the order of convergence of the primary error component quantitatively. The emergence of secondary higher-order error components was explained by the interaction of the discretisation error and the higher-order components of the manufactured source terms.

In the case of the steady manufactured solution a feature, which is common to all the results and which was not discussed before, is the slight checkerboard nature of the local error distributions. This can mostly be seen for the velocities. This is an indication of weak decoupling of the neighbouring values.

The skewness correction has a significant influence on the error around the local irregularities of a grid. The grids used in the verification studies consisted of mainly isotropic or near -isotropic triangles, for which the non-orthogonality and the associated error vanishes or is negligible. Thus, the error originating from such triangles is independent of the use of skewness correction. However, the results reveal that a large local error may have a significant non-local influence, especially in the case of an error in velocity, as the local error is convected with the flow into the other parts of a domain. Because of this, otherwise insignificant local irregularities in a grid can increase the error in a substantial part of the domain. As the error in this part depends on the severity of the irregularities rather than on the grid resolution, the local irregularities may also have an influence on the order of convergence. With first-order discretisation the overall level of error is so high that the influence of the grid irregularities is negligible in comparison, but with second-order discretisation the influence is clearly visible. In the first verification exercise

it was demonstrated that the influence of the local irregularities can be reduced significantly with skewness correction, even if the results also show that the influence cannot be removed entirely. The results also reveal that, compared to the Taylor extrapolation-based discretisation, the Frink discretisation for the convected velocity component is somewhat less sensitive to the irregularities.

One should also bear in mind that skewness correction is not used at the boundaries. Thus, a considerable error may originate from the grid irregularities at the boundaries, even if skewness correction is activated. Another typical observation related to the boundaries of a domain is the appearance of large errors in the corners, where a variable is extrapolated at both boundaries. The problems with the development of a disturbance at the ends of the free surface boundary are also believed to be related to this error in the corners. With most solver options this error converges at the same rate as the global error. However, with direct evaluation of the pressure term the order of convergence reduces rapidly with refinement. The large error is assumed to be related to the evaluation of the gradient combined with the sub-optimal quality of the triangulation in these corners. The error can be reduced by increasing the resolution of the triangulation towards the corners.

Part III

Conclusions

8 Concluding remarks and recommendations for future work

The substantial result of this work is a numerical method for the time-accurate simulation of free surface flows in two dimensions. It has been shown in Chapter 3 with a steady case that the robustness of the traditional fully decoupled free surface solution can be improved with the proposed approaches for the coupling of the flow and free surface solutions. The coupling has a particularly significant influence on the robustness of the method in case of the dynamic approach for free surface evaluation. However, the coupled kinematic approach has been observed to have superior convergence rate as well as better stability compared to the dynamic counterpart. The effectiveness of the coupling has also been demonstrated in case of an unsteady flow using the kinematic approach. However, in this case the influence of the coupling depends on the physical time step. With a time step that is large compared to the characteristic time scale of the flow the influence is significant, whereas the influence is insignificant with small time steps.

In Chapters 5 and 6 most of the discretisations presented in this work and as implemented in the solver have been verified as having the assumed, theoretical order of accuracy. This builds confidence in the implementation of the method, which is crucial from the point of view of the future use of the method as a research platform. In addition to the verification of the solution method, valuable information on the behaviour of the local numerical error with different discretisations has been obtained. In this regard, it has been observed that compared to the average level of error, there are systematically areas of larger error close to the boundaries – particularly at the corners. This suggests that the discretisations on the boundaries should be improved. The increased error in these areas is believed to be related at least partly to the evaluation of the gradients.

In addition to the discretisations, two different grid refinement strategies have also been studied in Chapter 5. The choice of the strategy has been shown to have an influence on the local distributions of error, but quite insignificant influence on the observed grid convergence. However, it should be noted that the geometry of the domain in the particular case has been very simple. Additional studies should be performed with more realistic grid geometries. In this respect particular attention should be paid on the refinement with grid regeneration. There are two reasons for this. First, it is assumed that the performance of the quadrisection refinement does not change significantly with the geometry

of the domain, as the refinement is guaranteed to give geometrically identical triangles in every case. Secondly, the quadrisection refinement is limited to a refinement factor $r = 2$. This may lead to prohibitively large grids if a comprehensive verification with a large set of grids is to be conducted. The refinement with grid regeneration is in this regard much more flexible and may, thus, be the preferred choice. Because of this, the possible issues related to the regeneration refinement should be explored in the future.

One of the main contributions of Chapter 6 is the demonstration of quantitative code verification of the surface tracking free surface solution approach. Often code verification of free surface solvers – if even performed – is based on an approximate form of the method of exact solutions. Usually some high order approximation of the solution of the non-linear free surface equations is used as a comparison. The proposed manufactured solution allows the verification of a time accurate surface tracking solution without any approximations.

Recently, independently of this work, Wang and Jia (2009) have proposed both steady and unsteady manufactured solutions for the code verification of free surface flow solvers. The solutions proposed by them differ from the one proposed in this work in that they do not have physical realism. Another fundamental difference lies in the satisfaction of the kinematic boundary condition. Their solutions satisfy the boundary condition, whereas the standing wave solution used in this work results in a non-zero source term. In addition to the proposal of the solutions, Wang and Jia encourage researchers to come up with their own manufactured solutions. The author of the thesis would like to concur with this encouragement.

The exponential-harmonic function and the corresponding simplification which have been constructed in the standing wave test case as approximations for the temporal variation of the primary wave component, have been shown to reproduce the variation relatively well in Chapter 6. The use of the approximation has extended the quantitative verification of the code from the use of global error norms taken over the full space-time domain into the use of the temporal variation of the primary error component for the verification. It has also been shown that with sufficiently small phase error and amplitude factor the additional simplification does not contribute significantly to the approximation error. In this case the primary source of approximation error is the exponential-harmonic construction itself. Because of the forcing, the altered equations, which include the manufactured source terms, represent the free propagation of a wave only approximately. This is a result of the linearisations which have been performed in the derivation of the standing wave so-

lution. However, this does not make the selected manufactured solution inferior in any respect. It should be remembered that it is not the purpose of the manufactured solution to satisfy the governing equations. If this would be the case, the method would revert to the method of exact solutions. However, because of the choice the detailed analysis has been somewhat complicated. The construction of a more accurate approximation for the temporal variation of the numerical solution which would take the forcing caused by the manufactured source terms into account, has been left outside the scope of this work and is recommended for future work. Another option is the use of a higher order solution for the standing wave. However, the forms of these may be overly complex (see e.g. Chen and Hsu, 2009), and the possible gains versus the required effort should be carefully assessed.

It should be noted that the findings on the behaviour of the numerical error presented in Chapter 6 are not entirely limited to standing waves. As mentioned in Sec. 6.3, a standing wave can be considered to be composed of two waves which travel in opposite directions. However, this is true only approximately, because of the nonlinear interaction of the wave components. Despite of this approximation, it is believed that the presented concepts and results are useful also in the analysis of the numerical behaviour of propagating waves. In the author's opinion, this topic is of increasing importance as volumetric discretisation based approaches are starting to complement model testing and traditional potential flow based methods in e.g. sea-keeping studies, in which it is fundamental that the conditions encountered by the ship are accurately reproduced.

References

Alessandrini, B. and Delhommeau, G., "A Fully Coupled Naiver-Stokes Solver for Calculation of Turbulent Incompressible Free Surface Flow Past a Ship Hull," *International Journal for Numerical Methods in Fluids*, Vol. 29, 1999, pp. 125–142.

Anderson, W.K., "Grid generation and flow solution method for Euler equations on unstructured grids," Technical memorandum NASA TM-4295, NASA Langley research center, 1992.

Apsley, D. and Hu, W., "CFD simulation of two- and three-dimensional free-surface flows," *International Journal for Numerical Methods in Fluids*, Vol. 42, 2003, pp. 465–491.

Barrett, R., Berry, M.W., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and van den Vorst, H., *Templates for the solution of linear systems: building blocks for iterative methods*, SIAM, 1993.

Barth, T.J. and Jespersen, D.C., "The design and application of upwind schemes on unstructured meshes," *AIAA 27th Aerospace Sciences Meeting*, Reno, Nevada, 1989, AIAA-89-0366.

Batina, J.T., "Unsteady Euler algorithm with unstructured dynamic mesh for Complex-Aircraft aerodynamic analysis," *AIAA Journal*, Vol. 29, No. 3, 1991, pp. 327–333.

Bet, F., Hänel, D., and Sharma, S.D., "Simulation of Hydrodynamical Free-Surface Flow," *ECCOMAS'96*, Elsevier, 1996.

Blom, F.J., "Considerations on the spring analogy," *International Journal for Numerical Methods in Fluids*, Vol. 32, 2000, pp. 647–668.

Burg, C. and Erwin, T., "Application of Richardson Extrapolation to the Numerical Solution of Partial Differential Equations," *Numerical Methods for Partial Differential Equations*, Vol. 25, 2009, pp. 810–832.

Burg, C.O. and Murali, V.K., "Efficient Code Verification Using the Residual Formulation of the Method of Manufactured Solutions," *34th AIAA Fluid Dynamics Conference and Exhibit*, Portland, USA, 2004, AIAA Paper 2004-2628.

Burg, C.O.E., Sreenivas, K., Hyams, D.G., and Mitchell, B., "Unstructured Nonlinear Free Surface Simulations for the Fully Appended DTMB Model 5415 Series Hull Includ-

ing Rotating Propulsors," *24th Symposium on Naval Hydrodynamics*, Fukuoka, Japan, 2002.

Cairncross, R.A., Schunk, P.R., Baer, T.A., Rao, R.R., and Sackinger, P.A., "A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion," *International Journal for Numerical Methods in Fluids*, Vol. 33, 2000, pp. 375–403.

Caretto, L.S., Gosman, A.D., Patankar, S.V., and Spalding, D.B., "Two calculation procedures for steady, three-dimensional flows with recirculation," *Proc. Third Int. Conf. Numer. Methods Fluid Dyn.*, Paris, 1972.

Carrica, P.M., Wilson, R.V., and Stern, F., "Unsteady RANS simulation of the ship forward speed diffraction problem," *Computers & Fluids*, Vol. 35, No. 6, 2006, pp. 545–570.

Chen, G., Kharif, C., Zaleski, S., and Li, J., "Two-dimensional Navier-Stokes simulation of breaking waves," *Physics of Fluids*, Vol. 11, No. 1, 1999, pp. 121–133.

Chen, Y.Y. and Hsu, H.C., "A third-order asymptotic solution of nonlinear standing water waves in Lagrangian coordinates," *Chinese Physics B*, Vol. 18, No. 3, 2009, pp. 861–871.

Davidson, L., "A pressure correction method for unstructured meshes with arbitrary control volumes," *International Journal for Numerical Methods in Fluids*, Vol. 22, 1996, pp. 265–281.

Demirdžić, I. and Muzaferija, S., "Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology," *Computer methods in applied mechanics and engineering*, Vol. 125, 1995, pp. 235–255.

Demirdžić, I., Muzaferija, S., and Perić, M., "Computation of turbulent flows in complex geometries, chap. 7," G.D. Tzabiras, ed., *Calculation of Complex Turbulent Flows*, WIT press, Southampton, U.K., 2000, pp. 249–299.

Deng, G.B., Quetey, P., and Visonneau, M., "A code verification exercise for the unstructured finite-volume solver ISIS-CFD," *ECCOMAS CFD 2006*, TU Delft, The Netherlands, 2006.

Donea, J., Huerta, A., Ponthot, J.P., and Rodríguez-Ferran, A., "Arbitrary Lagrangian-Eulerian Methods," E. Stein, R. de Borst, and T.J. Hughes, eds., *Encyclopedia of Computational Mechanics, Volume 1: Fundamentals*, John Wiley & Sons, Ltd., 2004, chap. 14.

Dulikravitch, G. and Huang, C., "Fast iterative algorithms based on optimized explicit time-stepping," *Computer Methods in Applied Mechanics and Engineering*, Vol. 63, 1987, pp. 15–36.

Duncan, J., "The Breaking and Non-Breaking Wave Resistance of a Two-Dimensional Hydrofoil," *Journal of Fluid Mechanics*, Vol. 126.

Eça, L. and Hoekstra, M., eds., 2^{nd} *Workshop on CFD Uncertainty Analysis*, Lisbon, Portugal, 2006a.

Eça, L. and Hoekstra, M., "On the influence of the iterative error in the numerical uncertainty of ship viscous flow calculations," 26^{th} *Symposium on Naval Hydrodynamics*, Rome, Italy, 2006b.

Eça, L. and Hoekstra, M., eds., 3^{rd} *Workshop on CFD Uncertainty Analysis*, Lisbon, Portugal, 2008.

Eça, L. and Hoekstra, M., "Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions," *Computers & Fluids*, Vol. 38, 2009, pp. 1580–1591.

Eça, L., Hoekstra, M., Hay, A., and Pelletier, D., "A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow," *7th world congress on computational mechanics*, Los Angeles, USA, 2006.

Eça, L., Hoekstra, M., Hay, A., and Pelletier, D., "Verification of RANS solvers with manufactured solutions," *Engineering with Computers*, Vol. 23, 2007, pp. 253–270.

Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., "Torsional springs for two-dimensional dynamic unstructured fluid meshes," *Computer methods in applied mechanics and engineering*, Vol. 163, 1998, pp. 231–245.

Farmer, J., "A Finite Volume Multigrid Solution to the Three Dimensional Nonlinear Ship Wave Problem," Ph.D. thesis, Princeton University, Department of Mechanical and Aerospace Engineering, 1993.

Ferziger, J.H. and Perić, M., *Computational Methods for Fluid Dynamics*, Springer, 1997, 2^{nd} ed.

Frink, N.T., "Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes," *AIAA Journal*, Vol. 30, No. 1, 1992, pp. 70–77.

Frink, N.T., "Recent Progress Towards a Three-Dimensional Unstructured Navier-Stokes Flow Solver," *AIAA 94-0061*, 32^{nd} Aerospace Sciences Meeting, Reno, Nevada.

Gnuplot, *gnuplot, An Interactive Plotting Program*, documentation for version 4.2 ed., 2007, available at `http://www.gnuplot.info/documentation.html`.

Golub, G. and van Loan, C., *Matrix computations*, Johns Hopkins University Press, Baltimore, 1990.

Hino, T., Martinelli, L., and Jameson, A., "A Finite-Volume Method with Unstructured Grid for Free Surface Flow Simulations," 6^{th} *International Conference on Numerical Ship Hydrodynamics*, Iowa, 1993.

Hino, T., ed., *CFD Workshop Tokyo 2005*, National Maritime Research Institute, Tokyo, Japan, 2005.

Hoekstra, M., Eça, L., Windt, J., and Raven, H., "Viscous flow calculations for KVLCC2 and KCS models using the PARNASSOS code," *Gothenburg 2000 A Workshop on Numerical Ship Hydrodynamics*, Gothenburg, Sweden, 2000.

Hoffren, J., "Unsteady Navier-Stokes Simulations of Airfoil Flows," C. Taylor, ed., *Numerical Methods in Laminar and Turbulent Flow*, Pineridge Press, 1993, Vol. VIII, pp. 1065–1076.

Hu, C. and Kashiwagi, M., "A CIP-based method for numerical simulations of violent free-surface flows," *Journal of Marine Science and Technology*, Vol. 9, 2004, pp. 143–157.

Hur, D.S., Kim, C.H., Kim, D.S., and Yoon, J.S., "Simulation of the nonlinear dynamic interactions between waves, a submerged breakwater and the seabed," *Ocean Engineering*, 2008, pp. 511–522.

Hänninen, S. and Mikkola, T., "On the implementation of an interface capturing method," V. Bertram, ed., 10^{th} *Numerical Towing Tank Symposium*, Hamburg, Germany, 2007.

ISO, *Guide to the Expression of Uncertainty in Measurement*, International Organization for Standardization, 1st ed., 1995.

ITTC, 19^{th} *International Towing Tank Conference*, 1990.

Jameson, A., Schmidt, W., and Turkel, E., "Numerical solutions of the Euler equations by finite volume methods with Runge-Kutta time stepping schemes," *AIAA 81-1259*.

Khosla, P.K. and Rubin, S.G., "A diagonally dominant second-order accurate implicit scheme," *Computers & Fluids*, Vol. 2, 1974, pp. 207–209.

Leroyer, A. and Visonneau, M., "Moving bodies in viscous flow simulation," V. Bertram, ed., 6^{th} *Numerical Towing Tank Symposium*, Rome, Italy, 2003.

Lomax, H., Pulliam, T.H., and Zingg, D.W., *Fundamentals of Computational Fluid Dynamics*, Springer Verlag, 2001.

Lungu, A. and Mori, K.h., "A Study on Numerical Schemes for More Accurate and Efficient Computations of Free-Surface Flows by Finite Difference Method," *Journal of Society of Naval Architects of Japan*, Vol. 173.

Löhner, R., Yang, C., Onate, E., and Idelssohn, S., "An unstructured grid-based parallel free surface flow solver," *Applied Numerical Mathematics*, Vol. 31, 1999, pp. 271–293.

Mavriplis, D. and Jameson, A., "Multigrid solution of the euler equations on unstructured and adaptive meshes," ICASE report 87-53, NASA Langley research center, 1987.

Mikkola, T., "Testing of Two FINFLO-based Free-surface Codes Using Eulerian Flow Over a Gaussian Ground Elevation," Internal report D-56, Helsinki University of Technology, Ship Laboratory, 1999.

Mikkola, T., "Implementation of an implicit scheme into a free surface RANS solver in order to improve the convergence," Tech. Rep. M-257, Helsinki University of Technology, Ship Laboratory, 2000, Master's Thesis.

Mikkola, T., "Development of an unstructured pressure-correction solver based on triangle meshes," G. Delhommeau and M. Visonneau, eds., 5^{th} *Numerical Towing Tank Symposium*, Ecole Centrale de Nantes, Pornichet, France, 2002.

Mikkola, T., "Numerical simulation of free surface flows in 2D with unstructured finite volume based pressure correction method," P. Råback, K. Santaoja, and R. Stenberg, eds., *VIII Finnish Mechanics Days*, Espoo, Finland, 2003.

Mikkola, T., "Time accurate simulation of free surface flows in 2D using unstructured grids," V. Bertram and H. Söding, eds., 7^{th} *Numerical Towing Tank Symposium*, Hamburg, Germany, 2004.

Mikkola, T., "On time accurate kinematic boundary condition in surface tracking context," V. Bertram and S. Kyulevcheliev, eds., 8^{th} *Numerical Towing Tank Symposium*, Varna, Bulgaria, 2005.

Mikkola, T., "Time Accurate Simulation of a Plunger Type Wave Maker Using Unstructured Finite Volume Solver with Surface Tracking," 26^{th} *Symposium on Naval Hydrodynamics*, Rome, Italy, 2006.

Mikkola, T., "Verification of a free surface code with method of manufactured solutions," V. Bertram, ed., 10^{th} *Numerical Towing Tank Symposium*, Hamburg, Germany, 2007.

Müller, J.D., "On triangles and flow," Ph.D. thesis, University of Michigan, 1996.

Murali, V.K. and Burg, C.O., "Verification of 2D Naiver-Stokes Codes by the Method of manufactured Solutions," *32nd AIAA Fluid Dynamics Conference and Exhibit*, St. Louis, USA, 2002, AIAA Paper 2002-3109.

Murthy, J.Y. and Mathur, S., "Periodic flow and heat transfer using unstructured meshes," *International Journal for Numerical Methods in Fluids*, Vol. 25, 1997, pp. 659–677.

Niceno, B., "EasyMesh," `http://www-dinma.univ.trieste.it/~nirftc/research/easymesh/,`.

Oberkampf, W.L., Blottner, F.G., and Aeschliman, D.P., "Methodology for Computational Fluid Dynamics Code Verification/Validation," *26th AIAA Fluid Dynamics Conference*, San Diego, USA, 1995.

Oberkampf, W.L. and Trucano, T.G., "Verification and validation in computational fluid dynamics," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 209–272.

Oberkampf, W.L., Trucano, T.G., and Hirsch, C., "Verification, validation, and predictive capability in computational engineering and physics," *Applied Mechanics Reviews*, Vol. 57, No. 5, 2004, pp. 345–384.

Oberkampf, W. and Blottner, F.G., "Issues in computational fluid dynamics: code verification and validation," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 687–695.

Patankar, S.V., *Numerical heat transfer and fluid flow*, McGraw-Hill, 1980.

Paterson, A., *A First Course in Fluid Dynamics*, Cambridge University Press, 1983.

Raven, H.C., van der Ploeg, A., and Starke, B., "Computation of free-surface viscous flows at model and full scale by a steady iterative approach," 25^{th} *Symposium on Naval Hydrodynamics*, St. John's, Canada, 2004.

Reed, H.L., Haynes, T.S., and Saric, W.S., "CFD validation issues in transition modelling," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 742–749.

Rhie, C.M. and Chow, W.L., "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation," *AIAA Journal*, Vol. 21, 1983, pp. 1525–1532.

Roache, P., "Quantification of uncertainty in computational fluid dynamics," *Annual Reviews in Fluid Mechanics*, Vol. 29, 1997, pp. 123–160.

Roache, P., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque NM, 1998.

Roache, P., "Code verification by the method of manufactured solutions," *Journal of Fluids Engineering*, Vol. 124.

Roy, C.J., Nelson, C.C., Smith, T.M., and Ober, C.C., "Verification of Euler/Navier-Stokes codes using the method of manufactured solutions," *International Journal for Numerical Methods in Fluids*, Vol. 44, 2004, pp. 599–620.

Roy, C.J., "Review of code and solution verification procedures for computational simulation," *Journal of Computational Physics*, Vol. 205, 2005, pp. 131–156.

Sackinger, P.A., Schunk, P.R., and Rao, R.R., "A Newton-Raphson Pseudo-Solid Domain Mapping Technique for Free and Moving Boundary Problems: A Finite Element Implementation," *Journal of Computational Physics*, Vol. 125, 1996, pp. 83–103.

Salari, K. and Knupp, P., "Code verification by the method of manufactured solutions," Tech. Rep. SAND2000-1444, Sandia National Laboratories, 2000.

Salari, K. and Knupp, P., *Verification of computer codes in computational science and engineering*, Chapman&Hall/CRC, 2003.

Schröder, P., "Subdivision as a fundamental building block of digital geometry processing algorithms," *Journal of Computational and Applied Mechanics*, Vol. 149, No. 1, 2002, pp. 207–219.

Schweighofer, J., "Investigation of two-dimensional transom waves using inviscid and viscous free-surface boundary conditions at model- and full-scale ship Reynolds numbers," Ph.D. thesis, Helsinki University of Technology, 2003.

Steinberg, S. and Roache, P., "Symbolic Manipulation and Computational Fluid Dynamics," *Journal of Computational Physics*, Vol. 57, No. 2.

Tahara, Y. and Stern, F., "A Large-Domain Approach for Calculating Ship Boundary Layers ans Wakes and Wave Fields for Nonzero Froude Number," *Journal of Computational Physics*, Vol. 127, 1996, pp. 398–411.

Thomadakis, M. and Leschziner, M., "A pressure-correction method for the solution of incompressible viscous flows on unstructured grids," *International Journal for Numerical Methods in Fluids*, Vol. 22, 1996, pp. 581–601.

Tzabiras, G.D., "A Numerical Investigation of 2D, Steady Free Surface Flows," *International Journal for Numerical Methods in Fluids*, Vol. 25, 1997, pp. 567–598.

van Brummelen, E.H., Raven, H.C., and Koren, B., "Efficient Numerical Solution of Steady Free-Surface Navier-Stokes Flow," *Journal of Computational Physics*, Vol. 174, 2001, pp. 120–137.

van Brummelen, E.H. and Segal, A., "Numerical solution of steady free-surface flows by the adjoint optimal shape design method," *International Journal for Numerical Methods in Fluids*, Vol. 41, 2003, pp. 3–27.

van den Vorst, H. and Sonneveld, P., "CGSTAB, a more smoothly converging variant of CGS," Tech. Rep. 90-50, Delft University of Technology, 1990.

van Doormal, J. and Raithby, G., "Enhancements of the SIMPLE method for predicting incompressible fluid flows," *Numerical Heat Transfer*, Vol. 7, 1984, pp. 147–163.

Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V: A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32.

Wang, J.W. and Liu, R.X., "A comparative study of finite volume methods on unstructured meshes for simulation of 2D shallow water wave problems," *Mathematics and Computers in Simulation*, Vol. 53, 2000, pp. 171–184.

Wang, S.S.Y. and Jia, Y., *Verification and Validation of 3D Free-Surface Flow Models*, ASCE Publications, 2009, chap. 4: Mathematical Verification Using Prescribed or Manufactured Solutions.

Wehausen, J.V. and Laitone, E.V., "Surface waves," *Encyclopedia of Physics*, Springer-Verlag, 1960, Vol. IX, pp. 446–778, available online at `http://www.coe.berkeley.edu/SurfaceWaves`.

Appendix A The complete results for the bump and the hydrofoil cases

This appendix presents in full the analysed results for the flow over a bump and the flow over a submerged hydrofoil which were discussed in Chapter 3. The analysis procedure was presented in Sec. 3.2. For each case the ratio of the L_2-norm of the changes at succesive iterations G_f and the assumed L_2-norm at the beginning of the iteration $|\Delta \tilde{f}^0|_2$ evaluated based on the dominant part of the convergence history are presented for the velocity components and the pressure. Additional, the average L_2-norm of the change for the converged solution $|\Delta \tilde{f}^\infty|_2$ is presented for each flow variable.

Table A.1: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a bump with the coupled kinematic approach (Case b). The lines with italic entries refer to cases, which failed to converge to the machine accuracy.

α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.5	.99932	5.4E-4	1.8E-7	.99933	4.0E-4	6.5E-8	.99933	4.2	3.6E-4
0.2	1.0	.99896	5.9E-4	1.9E-7	.99896	4.6E-4	6.7E-8	.99895	5.1	3.7E-4
0.2	1.3	.99878	6.6E-4	1.9E-7	.99879	4.9E-4	6.9E-8	.99878	5.5	3.7E-4
0.2	*1.4*	*.99869*	*7.6E-4*	*1.9E-7*	*.99870*	*5.6E-4*	*7.1E-8*	*.99870*	*6.3*	*3.8E-4*
0.3	0.5	.99913	6.4E-4	1.9E-7	.99913	5.1E-4	7.2E-8	.99913	3.6	4.3E-4
0.3	1.0	.99872	7.1E-4	1.9E-7	.99870	6.1E-4	7.5E-8	.99869	4.5	4.4E-4
0.3	1.1	.99863	7.9E-4	1.9E-7	.99863	6.1E-4	7.7E-8	.99862	4.6	4.4E-4
0.3	*1.2*	*.99857*	*7.9E-4*	*1.9E-7*	*.99856*	*6.3E-4*	*8.0E-8*	*.99856*	*4.8*	*4.5E-4*

Table A.2: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a bump with the dynamic approach (Case c). The lines with italic entries refer to cases, which failed to converge to the machine accuracy.

α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.5	.99932	3.5E-4	3.3E-7	.99939	1.5E-4	7.4E-8	.99936	2.1	3.7E-4
0.2	0.7	.99914	3.8E-4	3.0E-7	.99919	2.0E-4	7.1E-8	.99917	2.6	3.7E-4
0.2	*0.8*	*.99907*	*3.9E-4*	*2.9E-7*	*.99912*	*2.0E-4*	*7.2E-8*	*.99909*	*2.7*	*3.8E-4*
0.3	0.4	.99921	3.9E-4	3.6E-7	.99927	1.9E-4	8.4E-8	.99924	1.6	4.3E-4
0.3	*0.5*	*.99906*	*4.3E-4*	*3.3E-7*	*.99913*	*2.1E-4*	*8.1E-8*	*.99910*	*1.9*	*4.3E-4*

Table A.3: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a bump with the decoupled kinematic approach (Case d). The lines with italic entries refer to cases, which failed to converge to the machine accuracy.

α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.5	.99967	3.7E-4	1.9E-7	.99967	2.3E-4	6.7E-8	.99967	2.9	3.9E-4
0.2	1.0	.99950	6.4E-4	1.9E-7	.99949	4.9E-4	7.2E-8	.99949	5.5	4.2E-4
0.2	1.3	.99923	1.1E-3	1.9E-7	.99923	8.8E-4	7.7E-8	.99923	9.6	4.9E-4
0.2	*1.4*	*.99912*	*1.4E-3*	*2.1E-7*	*.99913*	*1.1E-3*	*1.1E-7*	*.99913*	*12.*	*7.6E-4*
0.3	0.5	.99952	5.6E-4	2.0E-7	.99952	3.6E-4	8.3E-8	.99952	2.9	5.4E-4
0.3	0.6	.99951	6.1E-4	2.0E-7	.99950	4.2E-4	8.4E-8	.99950	3.3	5.6E-4
0.3	*0.7*	*.99947*	*6.9E-4*	*2.4E-7*	*.99947*	*4.8E-4*	*1.5E-7*	*.99947*	*3.7*	*1.1E-3*

Table A.4: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a submerged hydrofoil with the coupled kinematic approach (Case b).

α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.3	.99832	9.0E-4	9.7E-8	.99831	9.0E-4	5.2E-8	.99834	7.9	5.3E-4
0.2	0.4	.99798	9.9E-4	9.6E-8	.99798	9.8E-4	5.1E-8	.99800	8.8	5.2E-4
0.2	0.5	.99764	1.0E-3	9.4E-8	.99762	1.0E-3	4.9E-8	.99767	9.0	4.9E-4
0.2	0.6	.99726	1.2E-3	9.4E-8	.99725	1.2E-3	4.9E-8	.99730	10.	4.9E-4
0.2	0.7	.99692	1.3E-3	9.5E-8	.99692	1.3E-3	4.9E-8	.99698	11.	5.0E-4
0.3	0.3	.99797	8.0E-4	9.6E-8	.99796	7.9E-4	5.0E-8	.99799	4.8	5.3E-4
0.3	0.4	.99757	7.3E-4	9.6E-8	.99757	7.2E-4	5.1E-8	.99761	4.3	5.4E-4
0.3	0.5	.99708	9.3E-4	9.6E-8	.99707	9.2E-4	5.0E-8	.99714	5.3	5.3E-4
0.3	0.6	.99654	1.3E-3	9.6E-8	.99654	1.2E-3	5.1E-8	.99664	6.9	5.3E-4
0.3	0.7	.99586	2.2E-3	9.6E-8	.99588	2.1E-3	5.1E-8	.99603	11.	5.3E-4

Table A.5: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a submerged hydrofoil with the coupled dynamic approach (Case c).

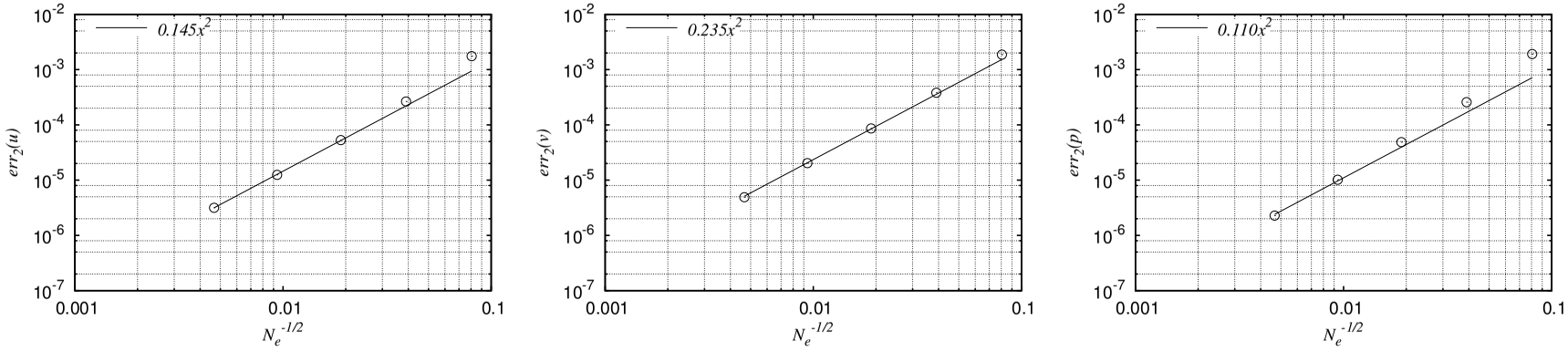
α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.3	.99876	8.5E-4	1.1E-7	.99876	7.1E-4	5.1E-8	.99878	6.0	5.0E-4
0.2	0.4	.99842	1.0E-3	1.1E-7	.99842	8.4E-4	5.1E-8	.99843	7.8	5.0E-4
0.2	0.5	.99814	1.1E-3	1.1E-7	.99813	1.0E-3	5.1E-8	.99812	9.8	4.9E-4
0.2	0.6	.99789	1.3E-3	1.1E-7	.99789	1.2E-3	5.1E-8	.99788	11.	5.1E-4
0.2	0.7	.99767	1.4E-3	1.1E-7	.99768	1.2E-3	5.1E-8	.99767	11.	4.9E-4
0.3	0.3	.99830	1.0E-3	1.1E-7	.99830	8.7E-4	5.3E-8	.99830	5.5	5.3E-4
0.3	0.4	.99787	1.3E-3	1.2E-7	.99788	1.1E-3	5.4E-8	.99788	7.0	5.3E-4
0.3	0.5	.99748	1.6E-3	1.1E-7	.99750	1.3E-3	5.3E-8	.99750	8.4	5.3E-4
0.3	0.6	.99719	1.7E-3	1.2E-7	.99721	1.4E-3	5.8E-8	.99722	8.6	5.4E-4
0.3	0.7	.99692	1.8E-3	1.4E-7	.99694	1.5E-3	7.7E-8	.99694	9.4	5.5E-4

Table A.6: The results of the analysis of the convergence histories of the velocity components and the pressure for the flow over a submerged hydrofoil with the decoupled kinematic approach (Case d).

α_p	$C_{\Delta\tau f}$	G_u	$\lvert\Delta\tilde{u}^0\rvert_2$	$\lvert\Delta\tilde{u}^\infty\rvert_2$	G_v	$\lvert\Delta\tilde{v}^0\rvert_2$	$\lvert\Delta\tilde{v}^\infty\rvert_2$	G_p	$\lvert\Delta\tilde{p}^0\rvert_2$	$\lvert\Delta\tilde{p}^\infty\rvert_2$
0.2	0.3	.99880	4.2E-4	9.4E-8	.99880	4.1E-4	4.9E-8	.99881	3.9	5.0E-4
0.2	0.4	.99859	4.2E-4	9.5E-8	.99859	4.1E-4	5.0E-8	.99861	3.9	5.1E-4
0.2	0.5	.99845	3.6E-4	9.5E-8	.99843	3.7E-4	4.9E-8	.99844	3.6	5.0E-4
0.2	0.6	.99824	3.7E-4	9.4E-8	.99822	3.9E-4	4.9E-8	.99823	3.8	5.0E-4
0.2	0.7	.99803	3.7E-4	9.4E-8	.99801	3.8E-4	4.9E-8	.99802	3.8	4.9E-4
0.3	0.3	.99850	3.9E-4	9.6E-8	.99849	3.9E-4	5.1E-8	.99851	2.5	5.3E-4
0.3	0.4	.99819	4.4E-4	9.6E-8	.99818	4.5E-4	5.1E-8	.99820	2.8	5.3E-4
0.3	0.5	.99797	3.8E-4	9.6E-8	.99795	3.9E-4	5.0E-8	.99797	2.5	5.3E-4
0.3	0.6	.99761	4.7E-4	9.7E-8	.99759	4.8E-4	5.0E-8	.99764	3.0	5.3E-4
0.3	0.7	.99731	5.2E-4	9.6E-8	.99731	5.2E-4	5.0E-8	.99737	3.2	5.3E-4

Appendix B The complete results for the steady manufactured solution

This appendix presents the full results separately for each of the steady manufactured solution cases, which were discussed in Chapter 5. For each case the L_2-norms of the absolute value of the error in the velocity components and the pressure are tabulated and presented as graphs. The corresponding observed orders of accuracy are also given. In addition to the error norms, the distributions of local error are presented with colour maps. No interpolation was used, and each control volume is coloured corresponding to the value at the centre of the volume. The maximum of each colour range is scaled according to the theoretical order of the discretisations (i.e. $p = 1$ or $p = 2$). It is twice the value given by the line which is fitted to the log-log graphs of the L_2-norm of error and has a slope equal to the theoretical order of accuracy. The line and the corresponding functional form are shown in the convergence graphs. It should be noted, that in this particular case the lines were not fitted to the data with a least-squares-type approach, but they were fitted by hand with visual inspection to highlight the convergence behaviour.



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
45978	3.15E-06	4.91E-06	2.28E-06	1.97	2.04	2.14	2.01	2.04	2.18	2.04	2.05	2.22
11402	1.25E-05	2.03E-05	1.02E-05	2.05	2.05	2.22	2.14	2.05	2.27	2.22	2.06	2.31
2782	5.29E-05	8.62E-05	4.87E-05	2.23	2.06	2.31	2.42	2.12	2.54	2.58	2.18	2.74
658	2.65E-04	3.81E-04	2.58E-04	2.60	2.19	2.76	-	-	-	-	-	-
154	1.75E-03	1.86E-03	1.91E-03	-	-	-	-	-	-	-	-	-

Figure B.1: Case 1: Set A, Taylor; order of accuracy

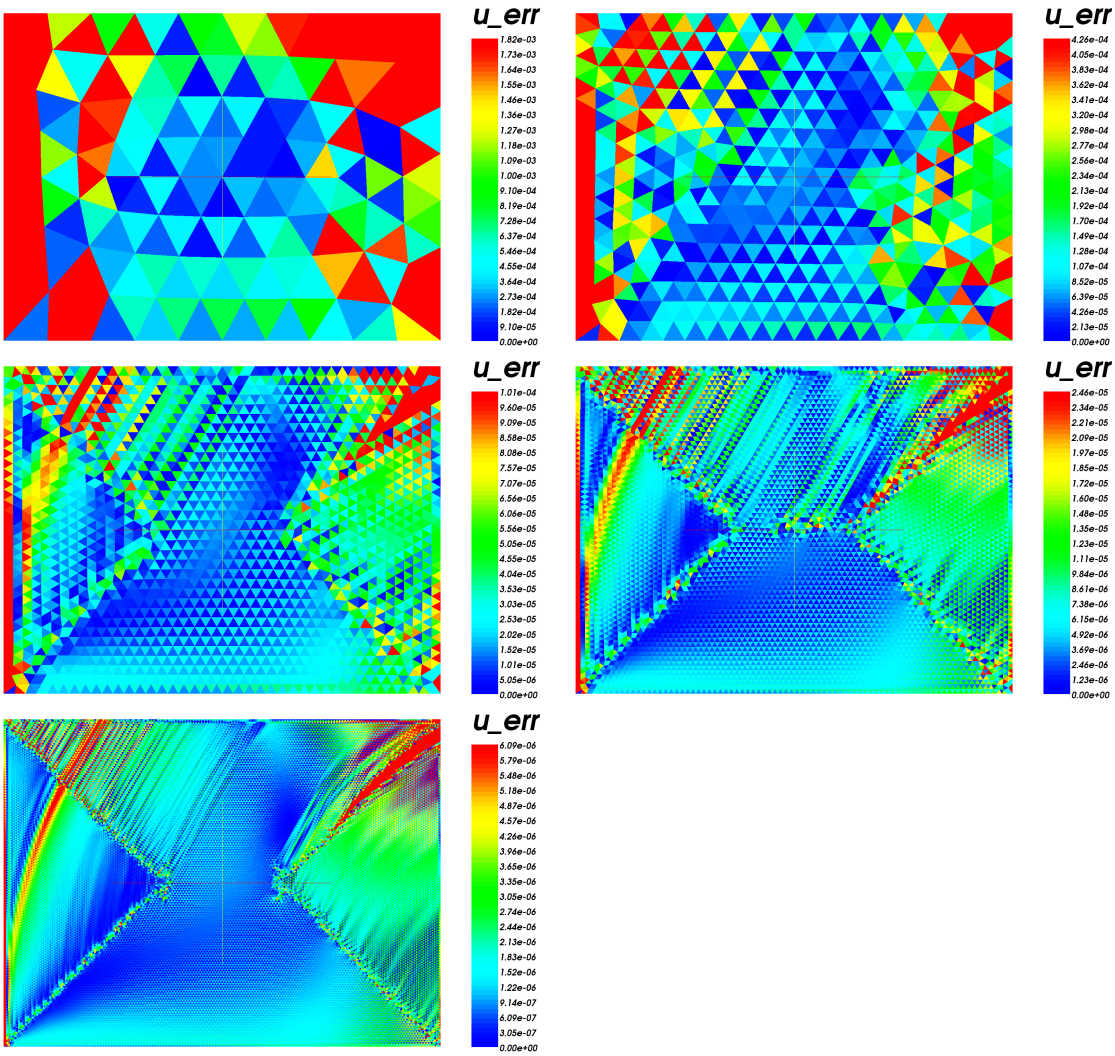


Figure B.2: Case 1: Set A, Taylor; x-velocity

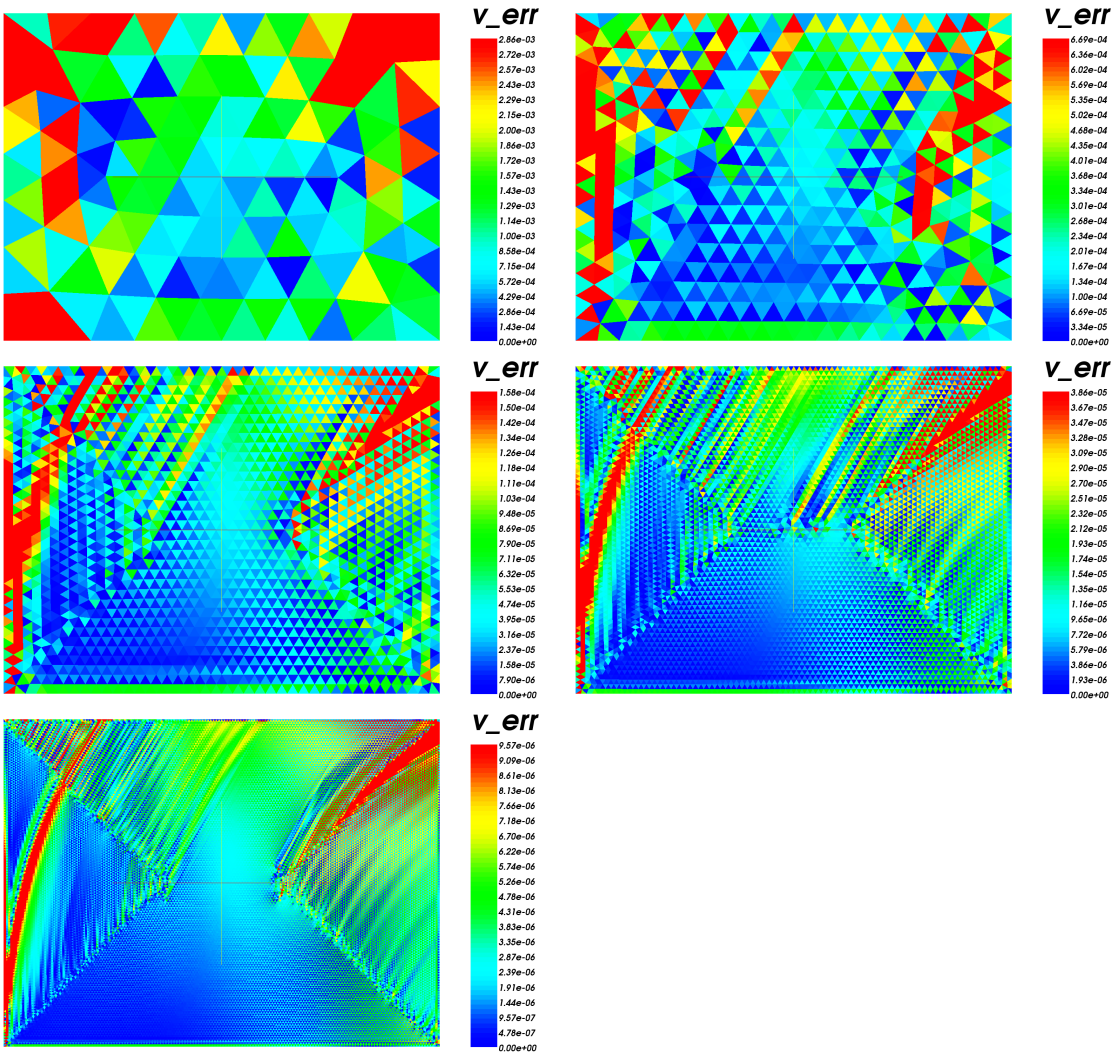


Figure B.3: Case 1: Set A, Taylor; y-velocity

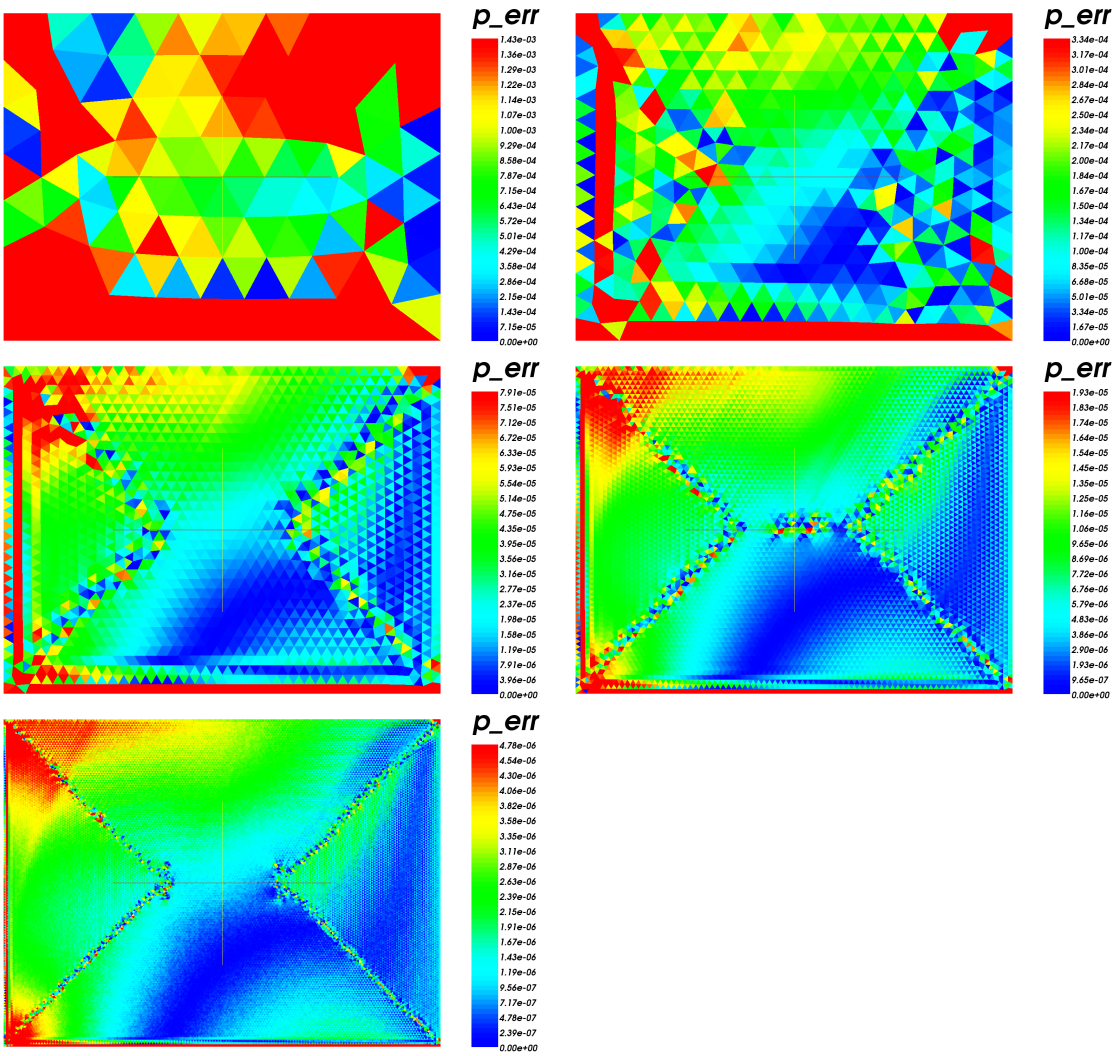
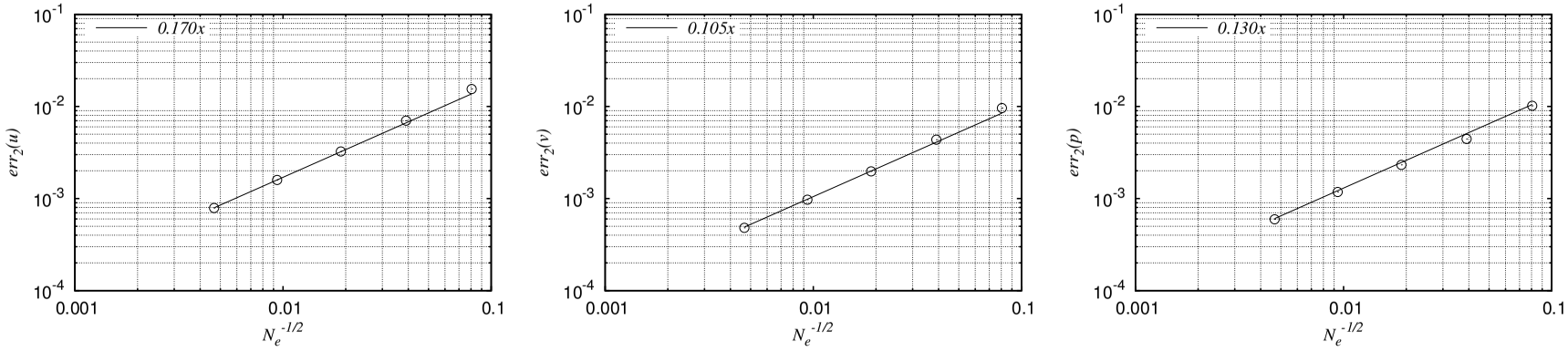


Figure B.4: Case 1: Set A, Taylor; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
45978	7.89E-04	4.81E-04	5.95E-04	1.01	1.01	0.98	1.01	1.01	0.97	1.01	1.01	0.97
11402	1.59E-03	9.71E-04	1.18E-03	1.01	1.01	0.96	1.04	1.05	0.93	1.05	1.07	0.92
2782	3.24E-03	1.97E-03	2.32E-03	1.07	1.10	0.90	1.08	1.09	1.02	1.08	1.09	1.08
658	7.01E-03	4.35E-03	4.44E-03	1.09	1.09	1.14	-	-	-	-	-	-
154	1.54E-02	9.62E-03	1.02E-02	-	-	-	-	-	-	-	-	-

Figure B.5: Case 2: Set A, first order; order of accuracy

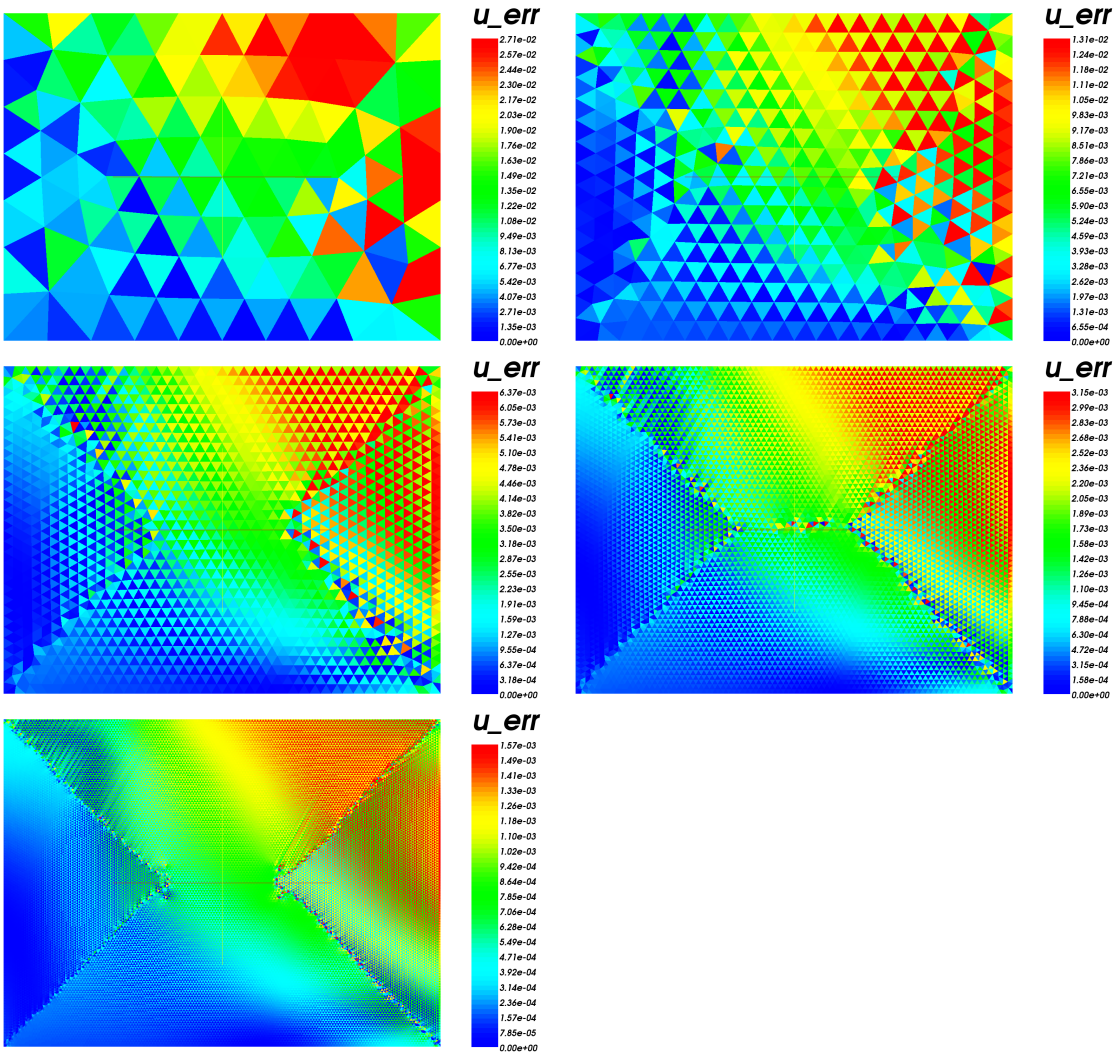


Figure B.6: Case 2: Set A, first order; x-velocity

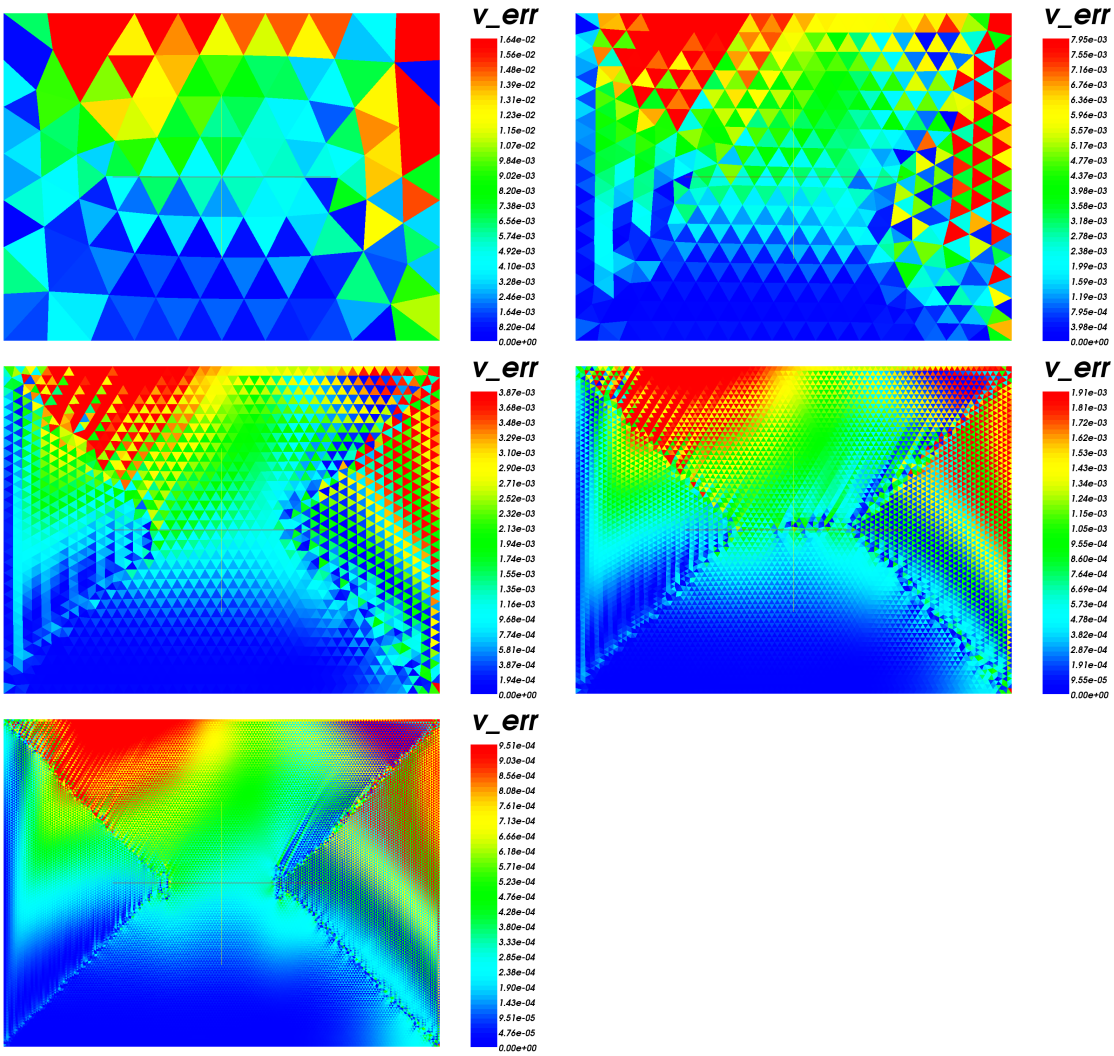


Figure B.7: Case 2: Set A, first order; y-velocity

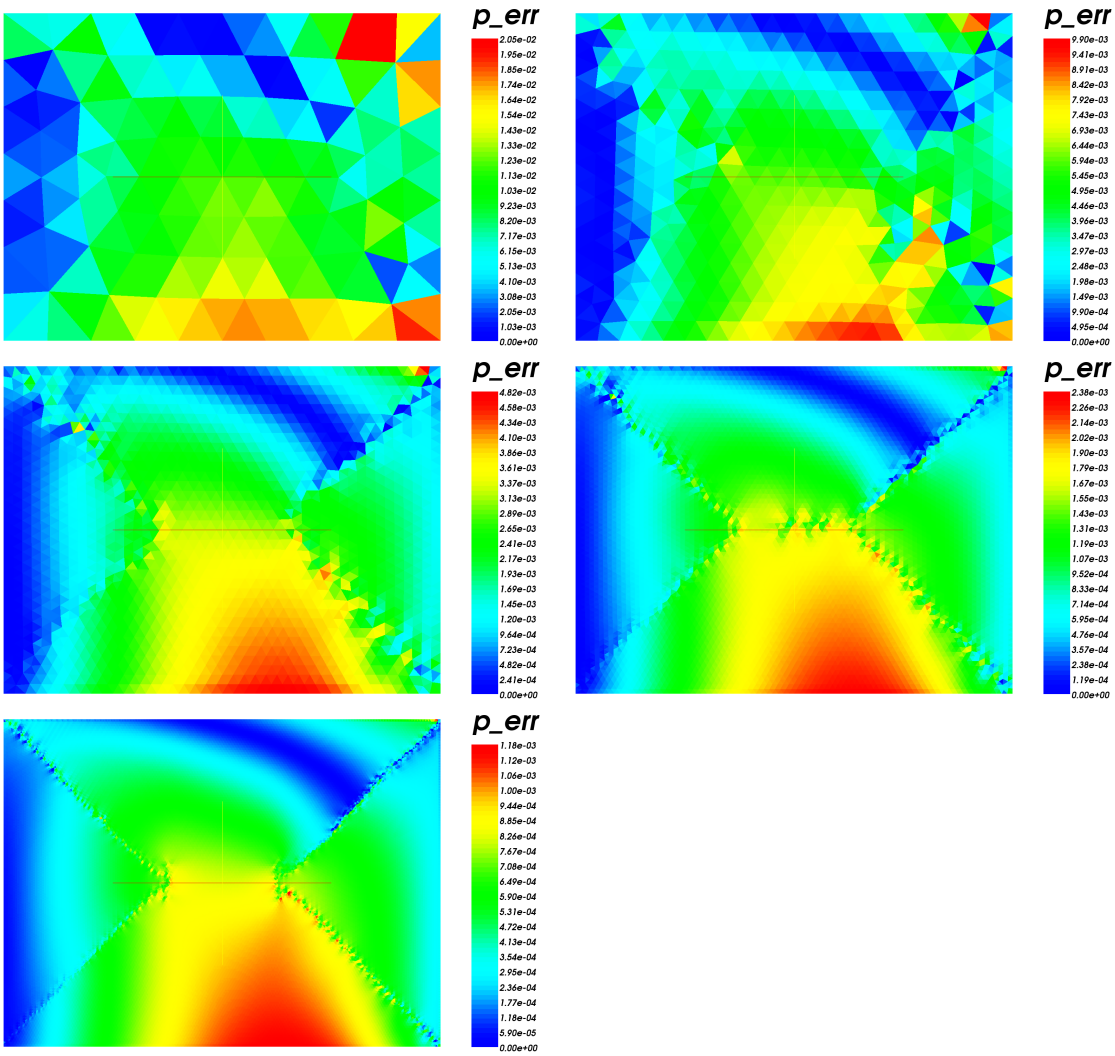
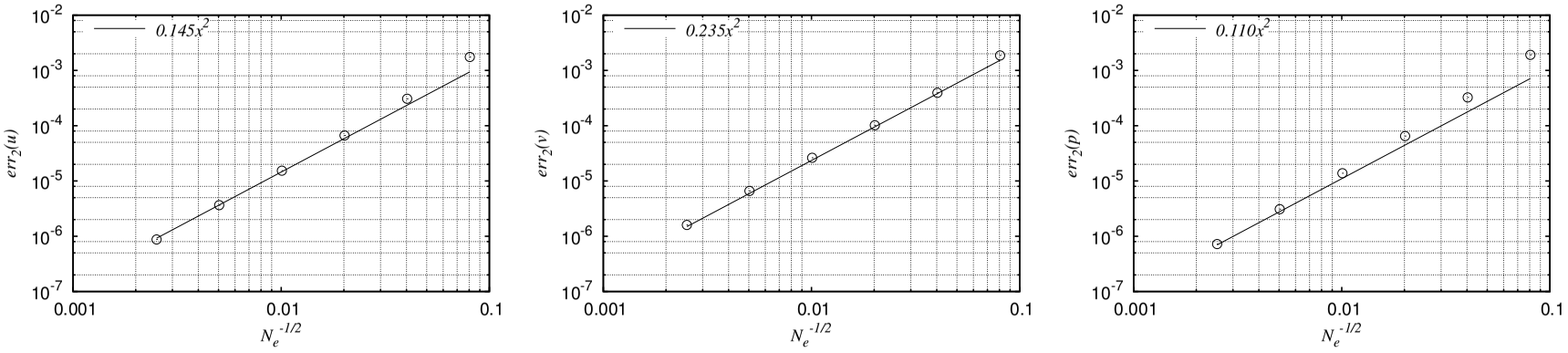


Figure B.8: Case 2: Set A, first order; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	8.74E-07	1.60E-06	7.23E-07	2.06	2.04	2.10	2.07	2.02	2.13	2.07	2.00	2.15
39424	3.65E-06	6.60E-06	3.09E-06	2.07	1.99	2.16	2.10	1.97	2.20	2.11	1.96	2.23
9856	1.54E-05	2.63E-05	1.38E-05	2.12	1.95	2.24	2.16	1.95	2.28	2.20	1.95	2.31
2464	6.66E-05	1.02E-04	6.50E-05	2.20	1.95	2.32	2.36	2.10	2.44	2.49	2.22	2.55
616	3.07E-04	3.94E-04	3.25E-04	2.51	2.24	2.56	-	-	-	-	-	-
154	1.75E-03	1.86E-03	1.91E-03	-	-	-	-	-	-	-	-	-

Figure B.9: Case 3: Set B, Taylor; order of accuracy

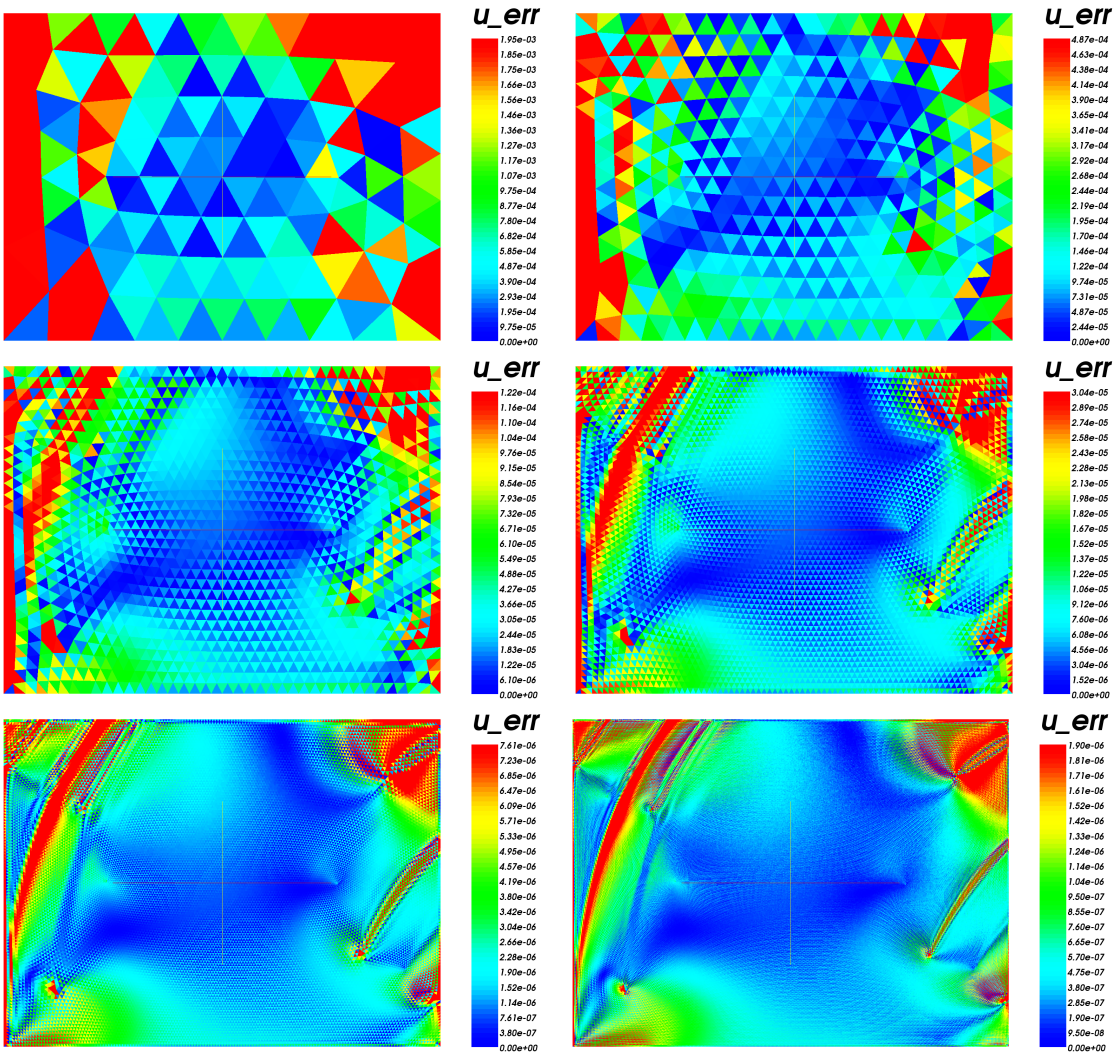


Figure B.10: Case 3: Set B, Taylor; x-velocity

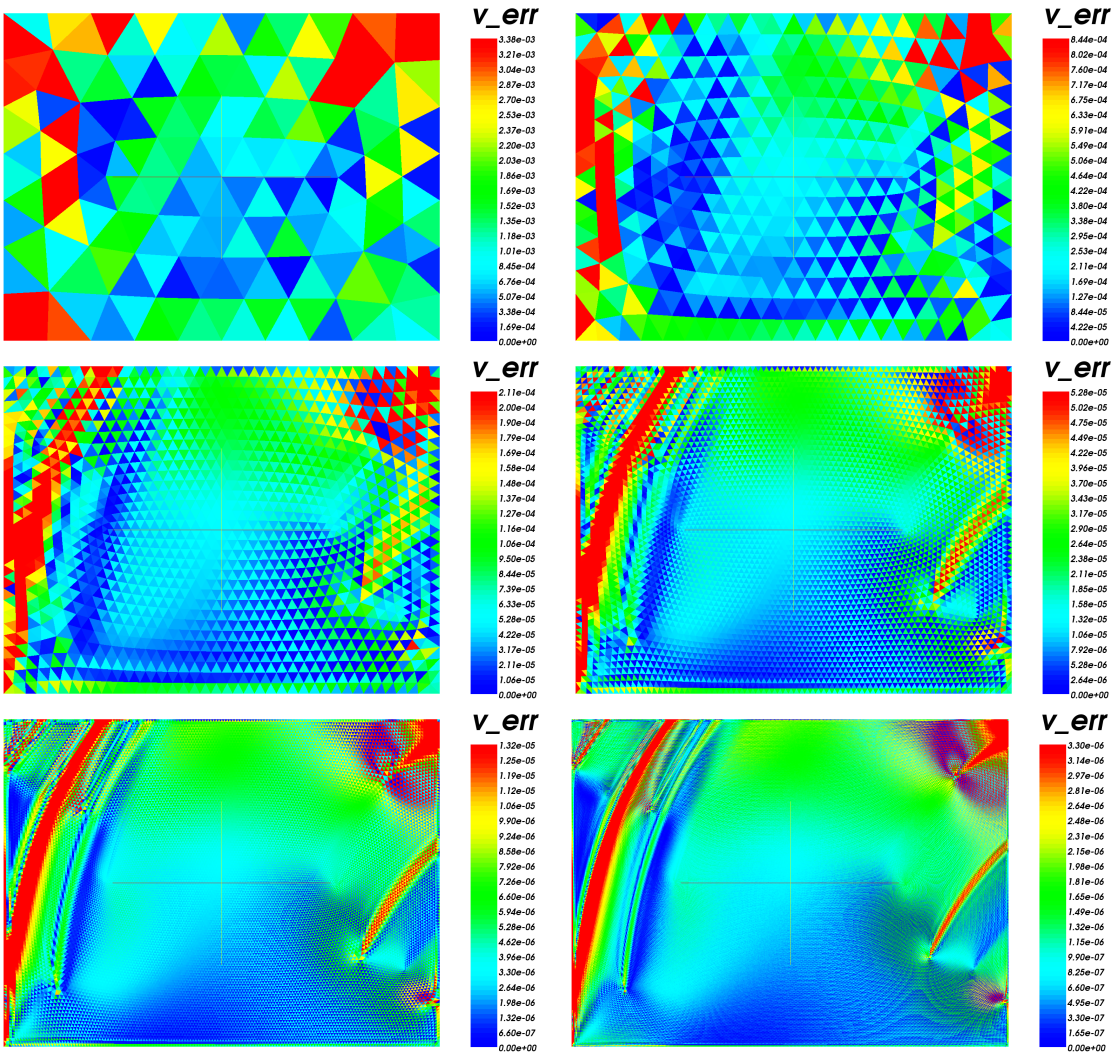


Figure B.11: Case 3: Set B, Taylor; y-velocity

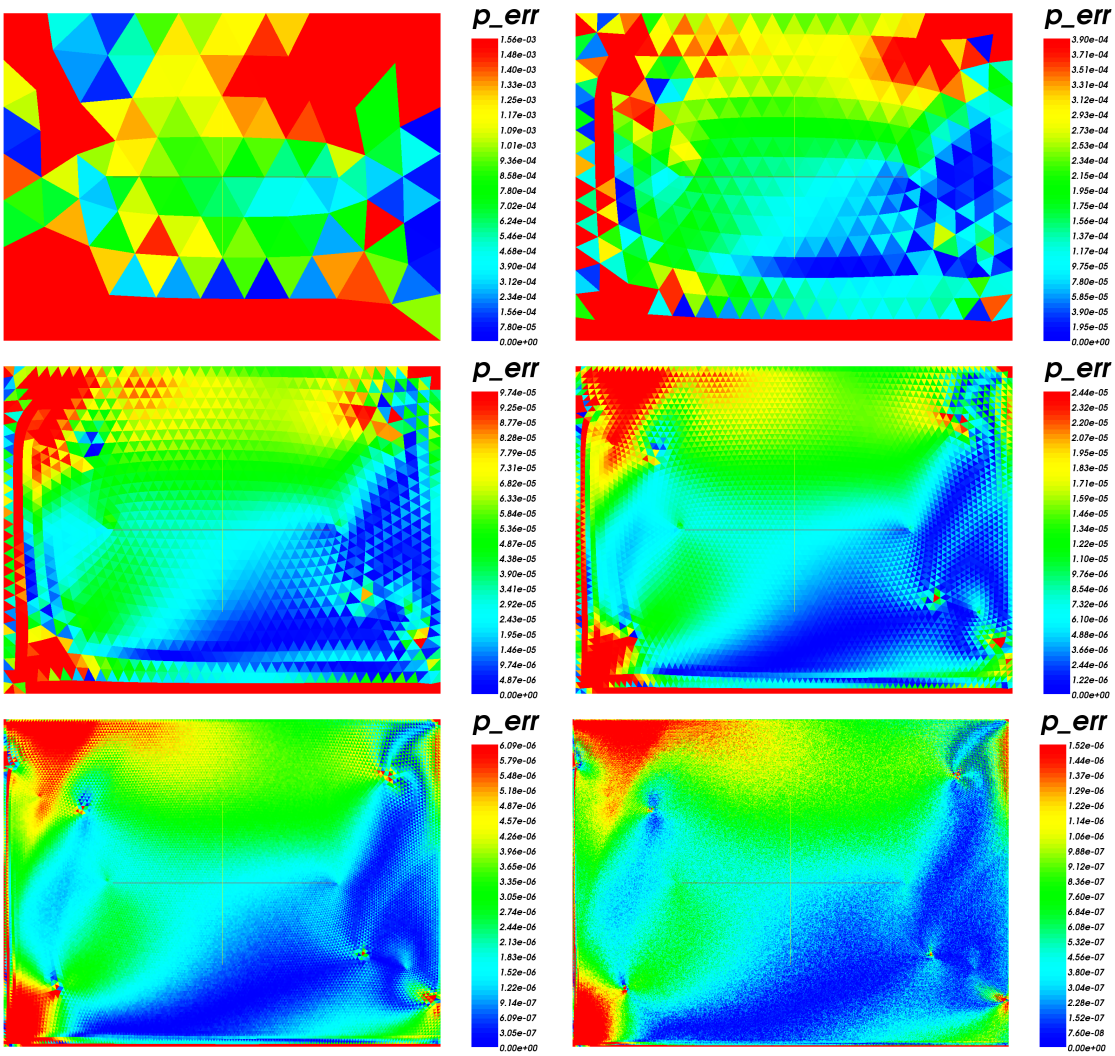
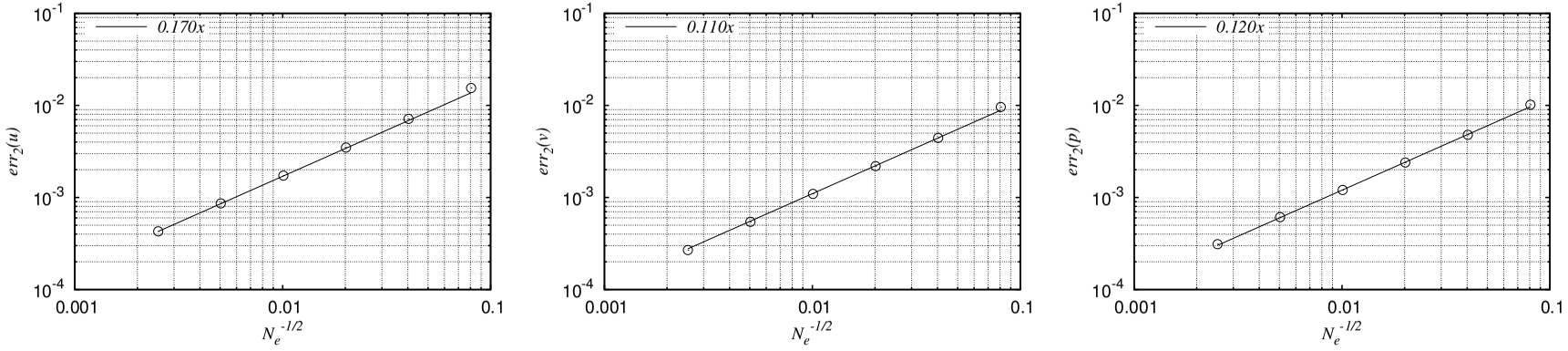


Figure B.12: Case 3: Set B, Taylor; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	4.29E-04	2.69E-04	3.11E-04	1.01	1.02	0.97	1.01	1.01	0.98	1.00	1.01	0.98
39424	8.63E-04	5.44E-04	6.11E-04	1.00	1.01	0.98	1.01	1.00	0.99	1.01	1.00	0.99
9856	1.73E-03	1.09E-03	1.21E-03	1.01	1.00	0.99	1.02	1.01	0.99	1.03	1.02	1.00
2464	3.48E-03	2.19E-03	2.40E-03	1.04	1.02	1.00	1.07	1.07	1.04	1.09	1.09	1.06
616	7.14E-03	4.44E-03	4.79E-03	1.11	1.12	1.09	-	-	-	-	-	-
154	1.54E-02	9.62E-03	1.02E-02	-	-	-	-	-	-	-	-	-

Figure B.13: Case 4: Set B, first-order; order of accuracy

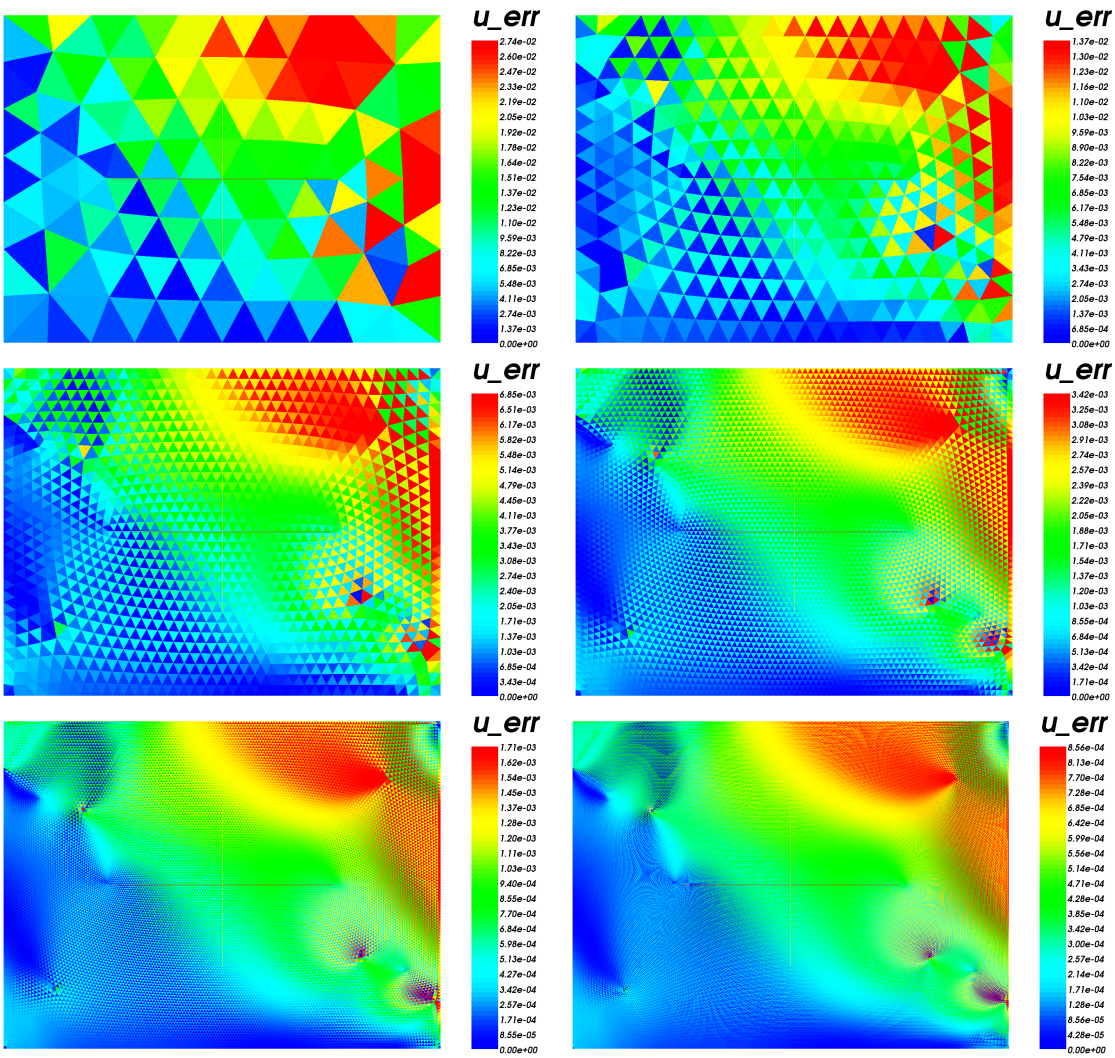


Figure B.14: Case 4: Set B, first-order; x-velocity

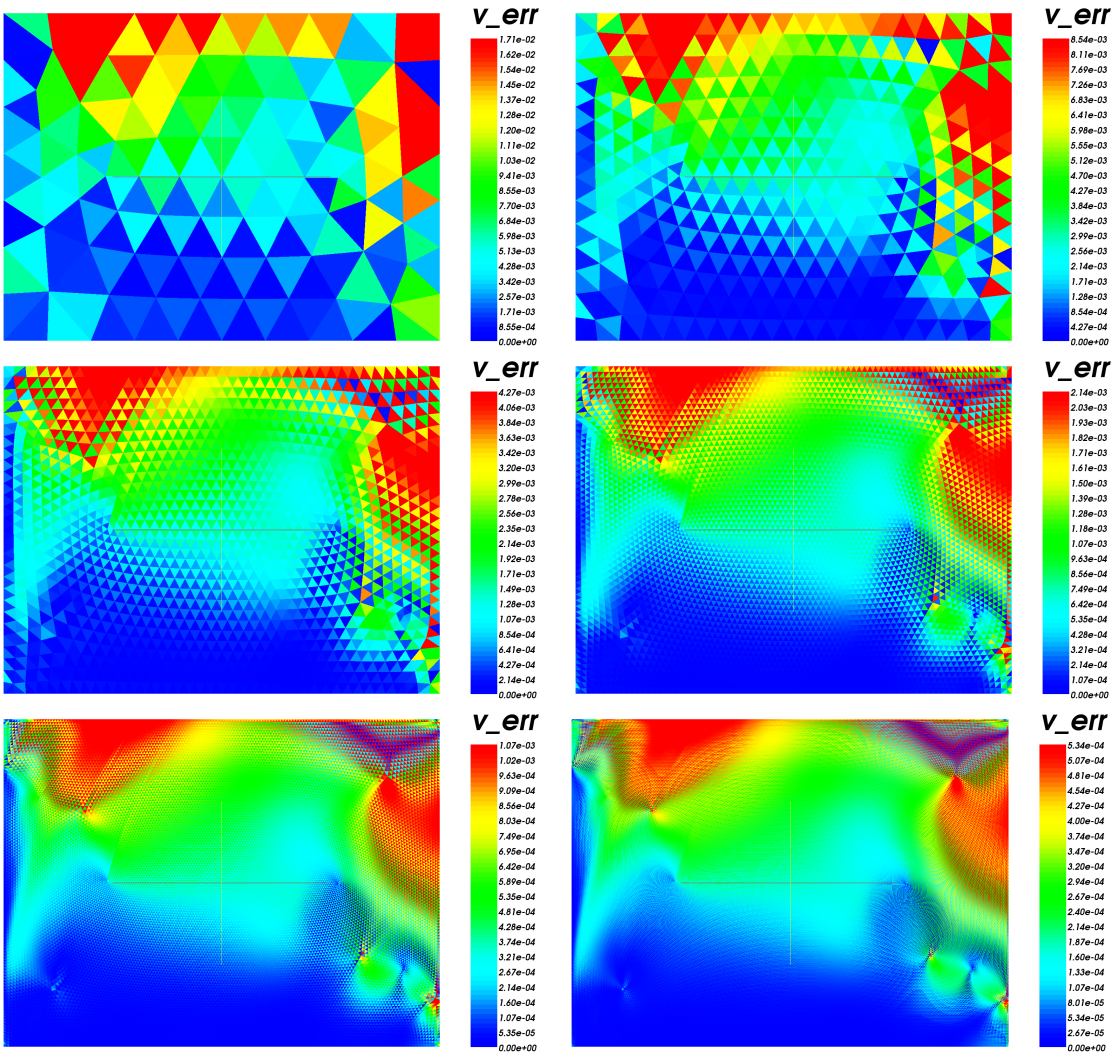


Figure B.15: Case 4: Set B, first-order; y-velocity

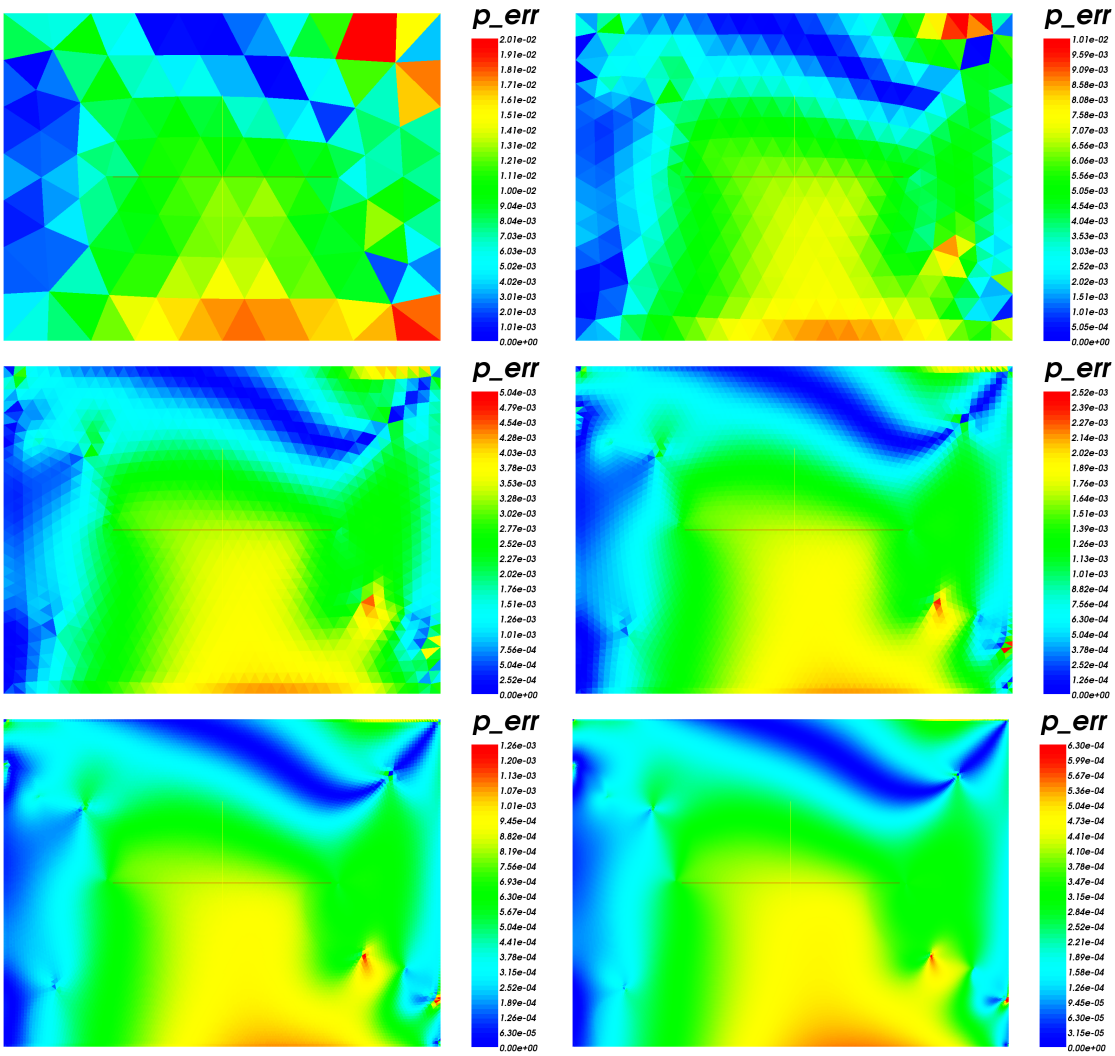
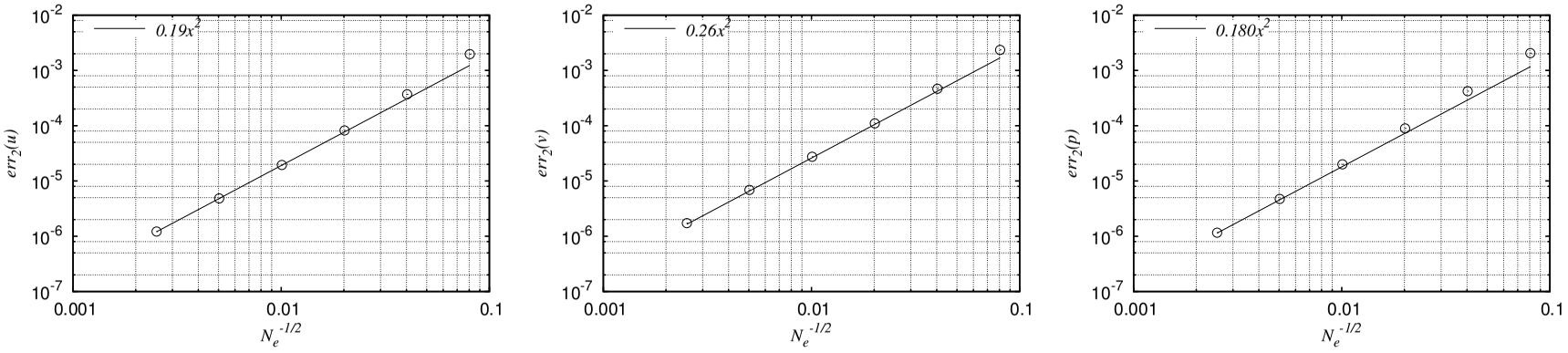


Figure B.16: Case 4: Set B, first-order; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	1.22E-06	1.73E-06	1.16E-06	1.99	2.00	2.03	2.00	1.99	2.05	2.01	1.99	2.08
39424	4.83E-06	6.89E-06	4.72E-06	2.01	1.99	2.08	2.04	2.00	2.12	2.07	2.00	2.15
9856	1.95E-05	2.74E-05	2.00E-05	2.07	2.00	2.16	2.12	2.04	2.20	2.16	2.07	2.23
2464	8.21E-05	1.10E-04	8.93E-05	2.17	2.08	2.24	2.29	2.21	2.26	2.40	2.32	2.28
616	3.69E-04	4.65E-04	4.22E-04	2.41	2.33	2.28	-	-	-	-	-	-
154	1.97E-03	2.35E-03	2.05E-03	-	-	-	-	-	-	-	-	-

Figure B.17: Case 5: Set B, Frink; order of accuracy

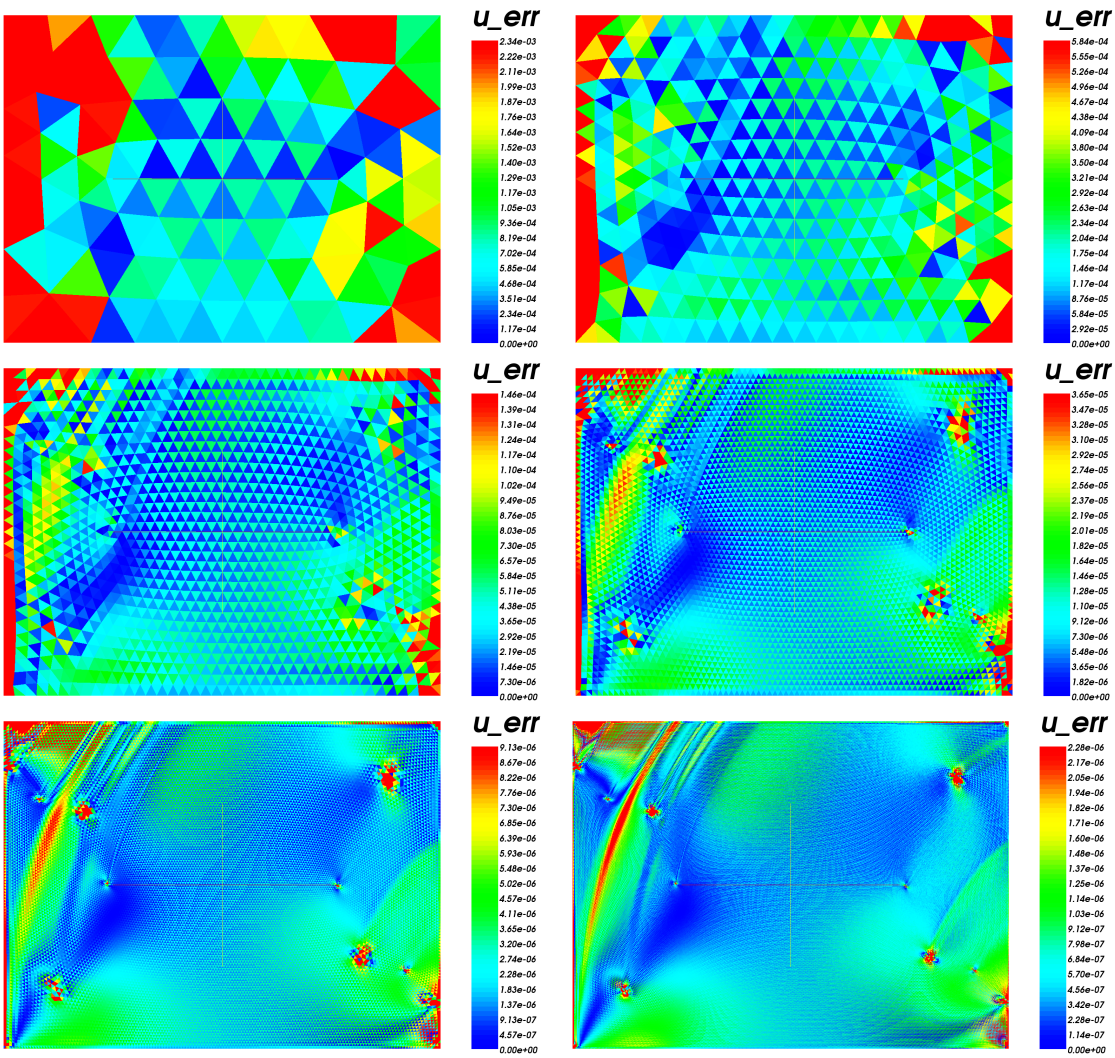


Figure B.18: Case 5: Set B, Frink; x-velocity

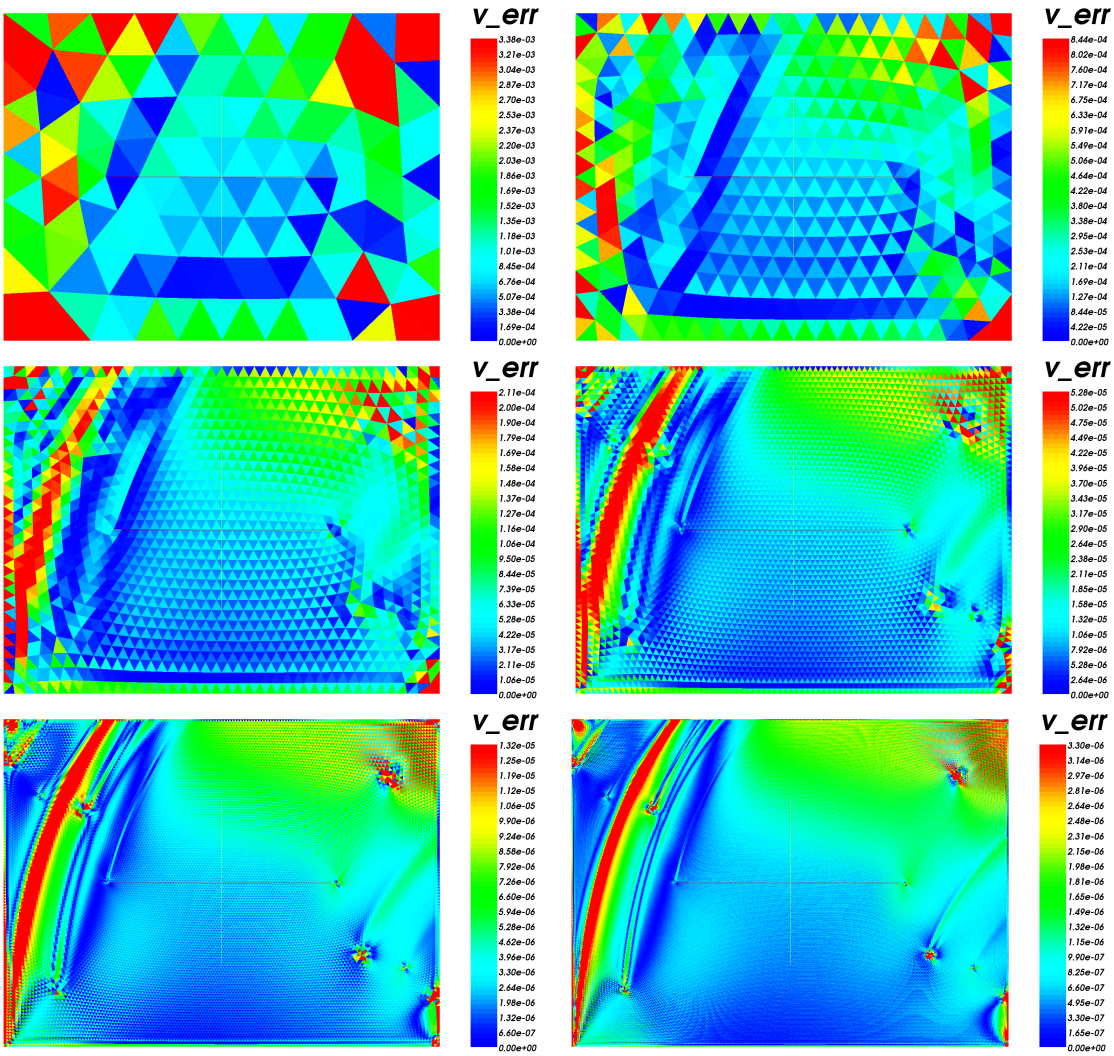


Figure B.19: Case 5: Set B, Frink; y-velocity

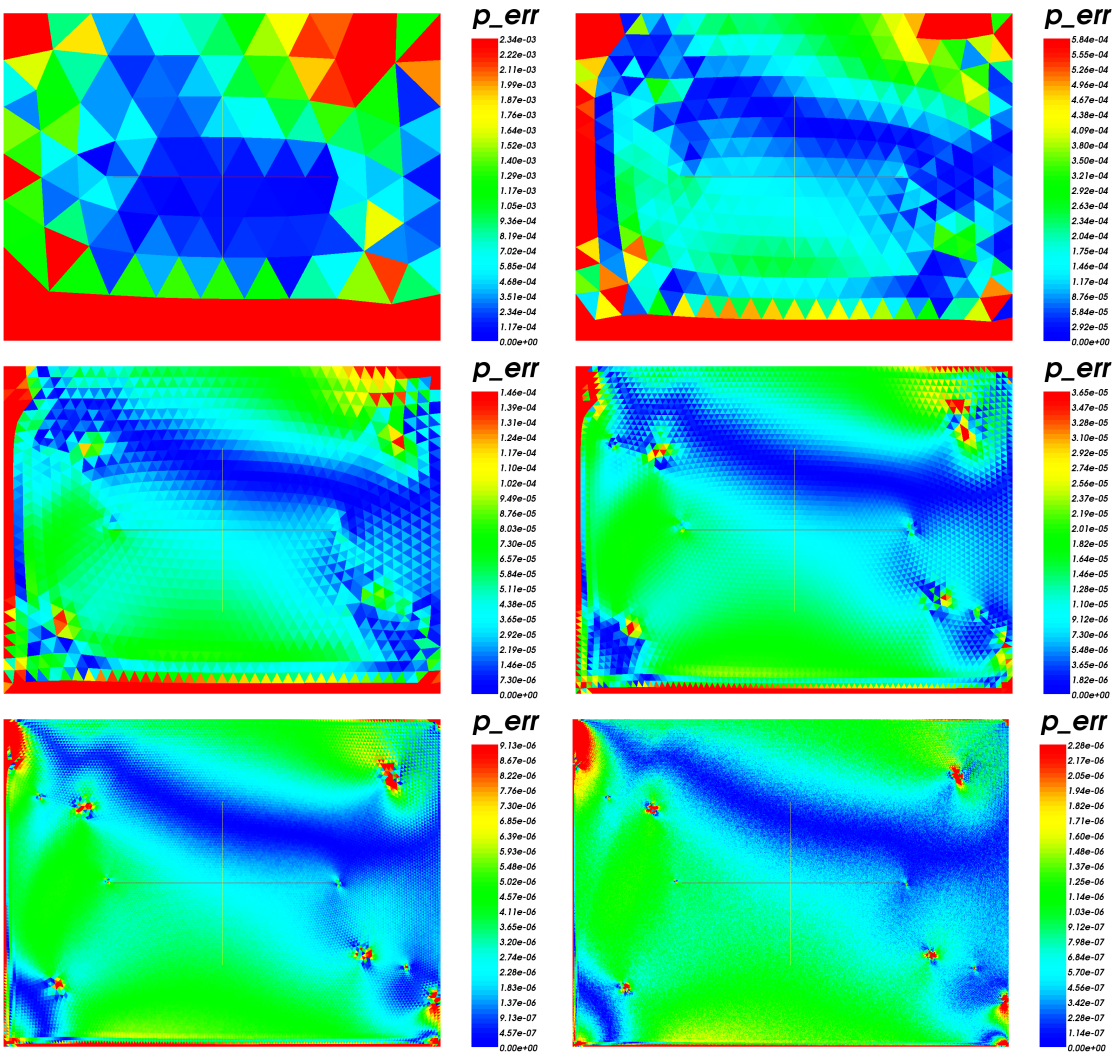
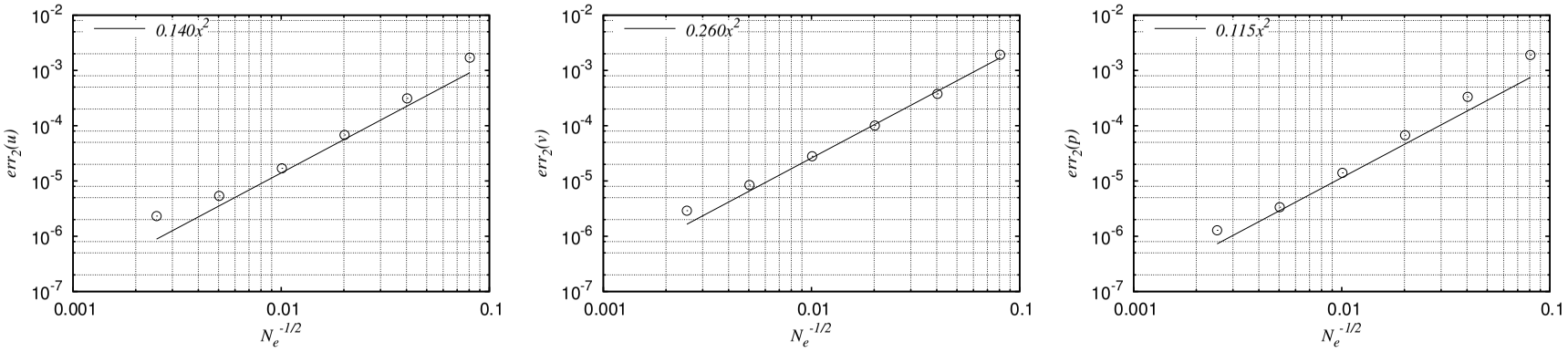


Figure B.20: Case 5: Set B, Frink; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	2.32E-06	2.90E-06	1.28E-06	1.21	1.54	1.39	1.43	1.64	1.73	1.58	1.70	1.98
39424	5.38E-06	8.45E-06	3.35E-06	1.66	1.73	2.07	1.83	1.79	2.16	1.96	1.83	2.25
9856	1.69E-05	2.80E-05	1.40E-05	2.00	1.84	2.26	2.10	1.87	2.28	2.18	1.90	2.30
2464	6.79E-05	1.00E-04	6.73E-05	2.20	1.91	2.31	2.32	2.13	2.41	2.43	2.31	2.50
616	3.12E-04	3.77E-04	3.33E-04	2.45	2.35	2.51	-	-	-	-	-	-
154	1.70E-03	1.91E-03	1.90E-03	-	-	-	-	-	-	-	-	-

Figure B.21: Case 6: Set B, pressure term with gradient; order of accuracy

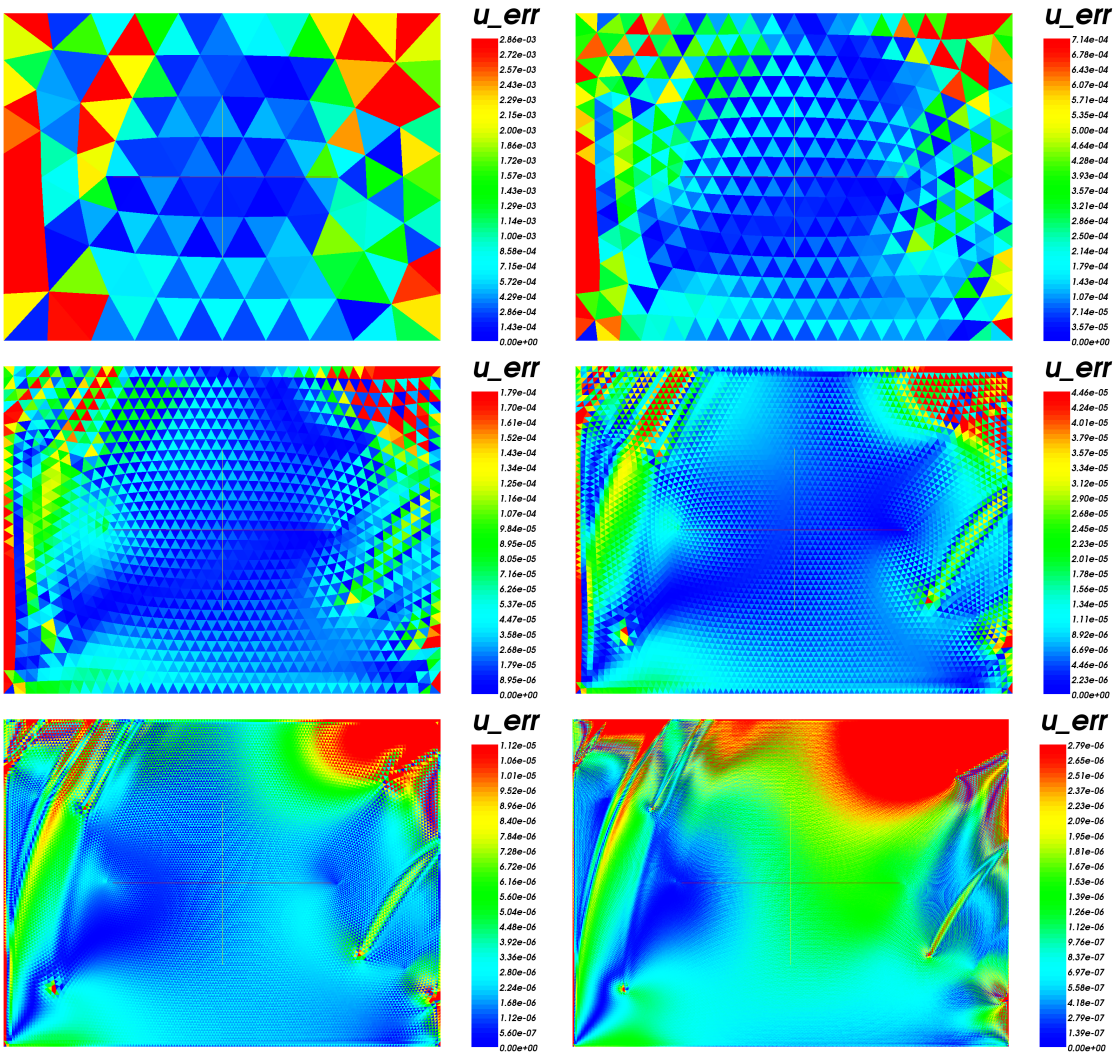


Figure B.22: Case 6: Set B, pressure term with gradient; x-velocity

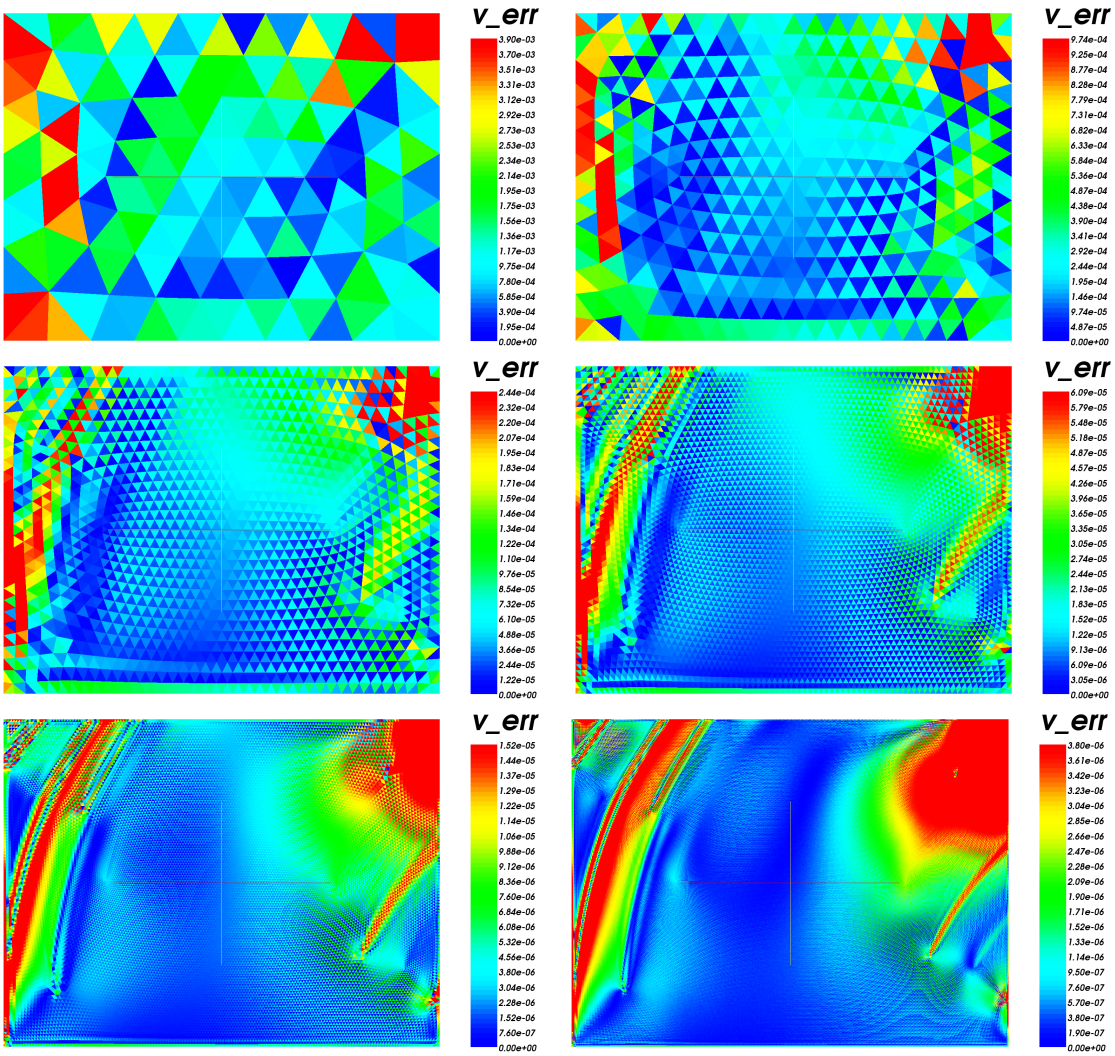


Figure B.23: Case 6: Set B, pressure term with gradient; y-velocity

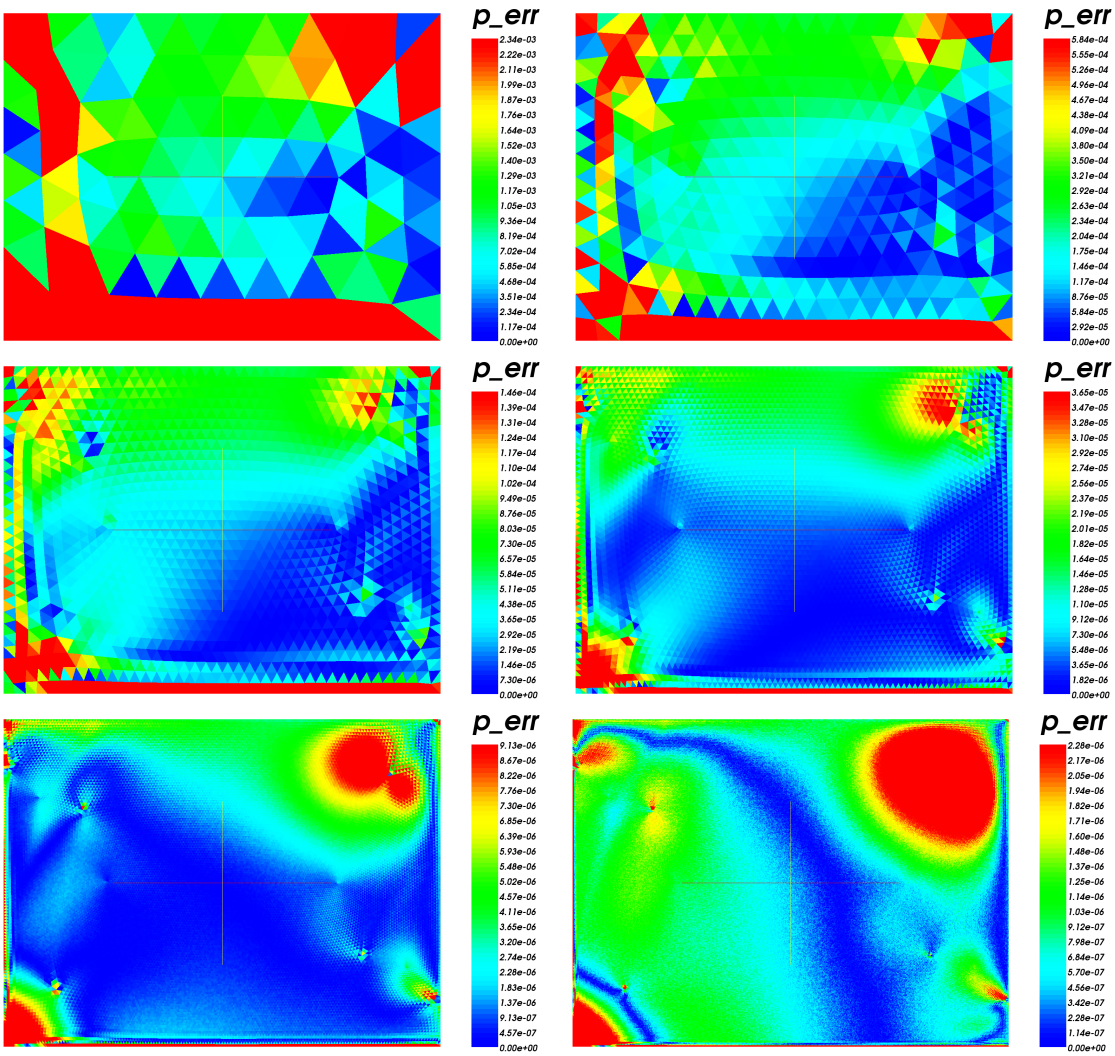
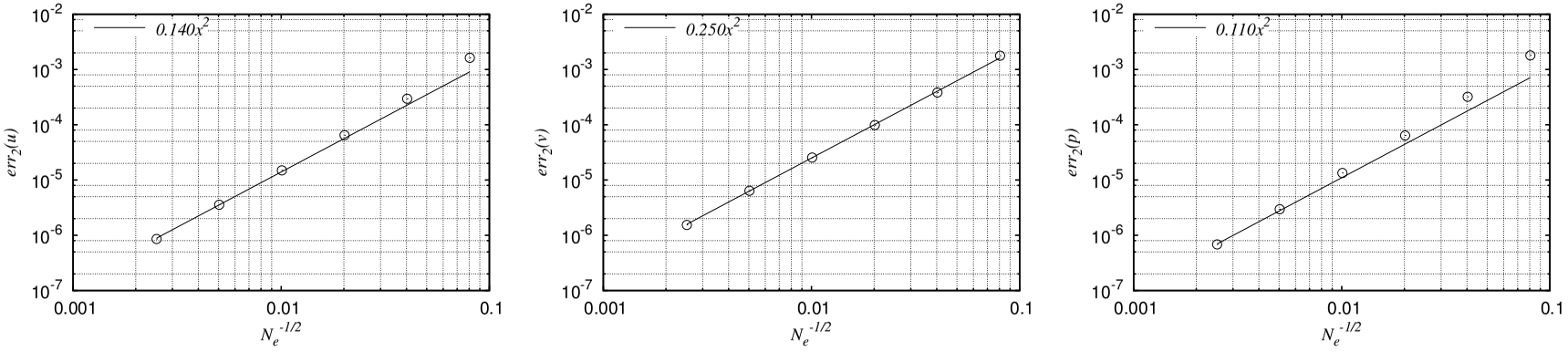


Figure B.24: Case 6: Set B, pressure term with gradient; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	8.54E-07	1.54E-06	6.83E-07	2.06	2.05	2.12	2.06	2.03	2.15	2.07	2.00	2.17
39424	3.56E-06	6.38E-06	2.96E-06	2.07	2.00	2.18	2.09	1.98	2.22	2.11	1.96	2.25
9856	1.50E-05	2.55E-05	1.34E-05	2.11	1.95	2.25	2.15	1.96	2.29	2.18	1.96	2.32
2464	6.47E-05	9.88E-05	6.39E-05	2.19	1.96	2.32	2.33	2.08	2.41	2.45	2.19	2.48
616	2.95E-04	3.84E-04	3.20E-04	2.46	2.21	2.49	-	-	-	-	-	-
154	1.63E-03	1.78E-03	1.80E-03	-	-	-	-	-	-	-	-	-

Figure B.25: Case 7: Set B, gradients with Gauss integration; order of accuracy

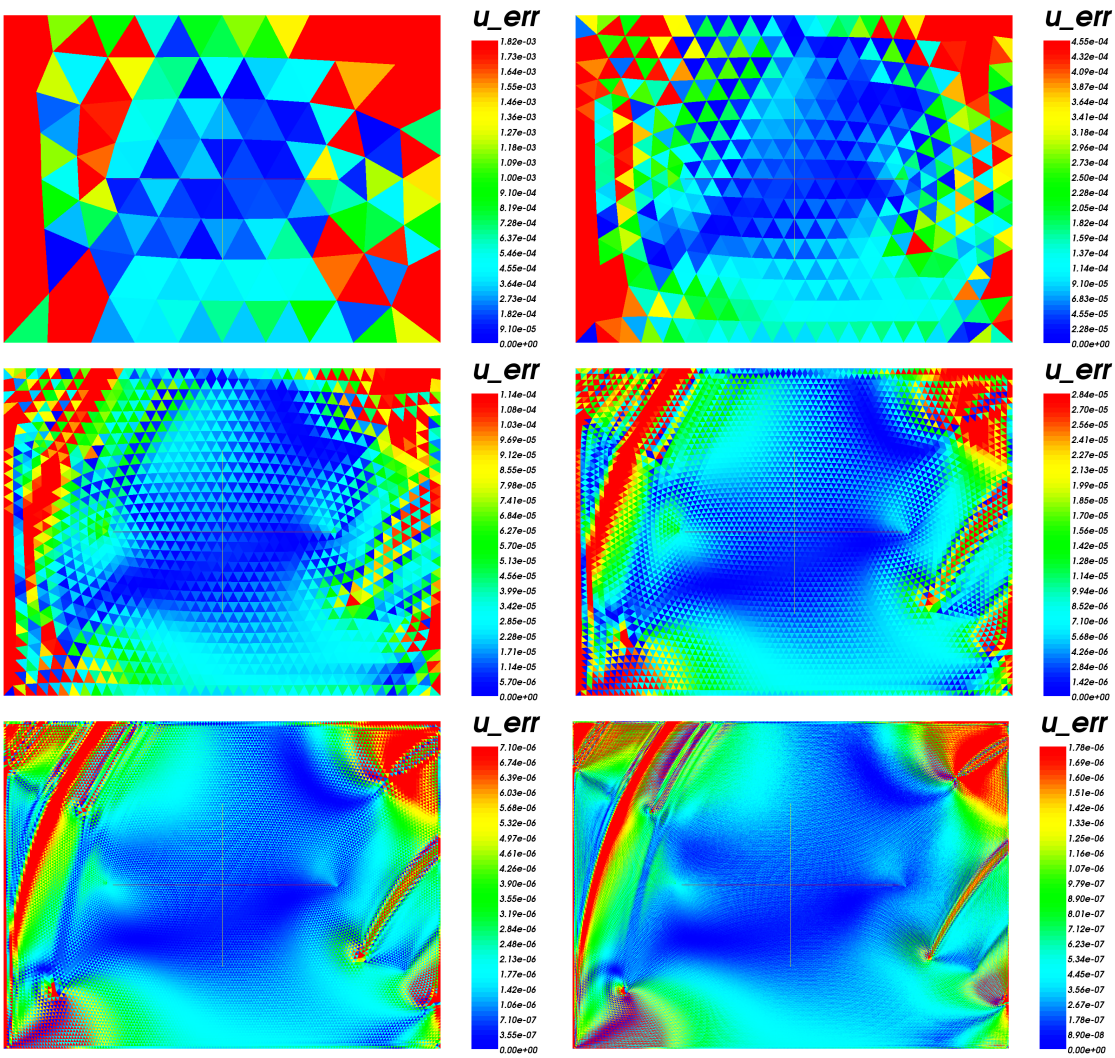


Figure B.26: Case 7: Set B, gradients with Gauss integration; x-velocity

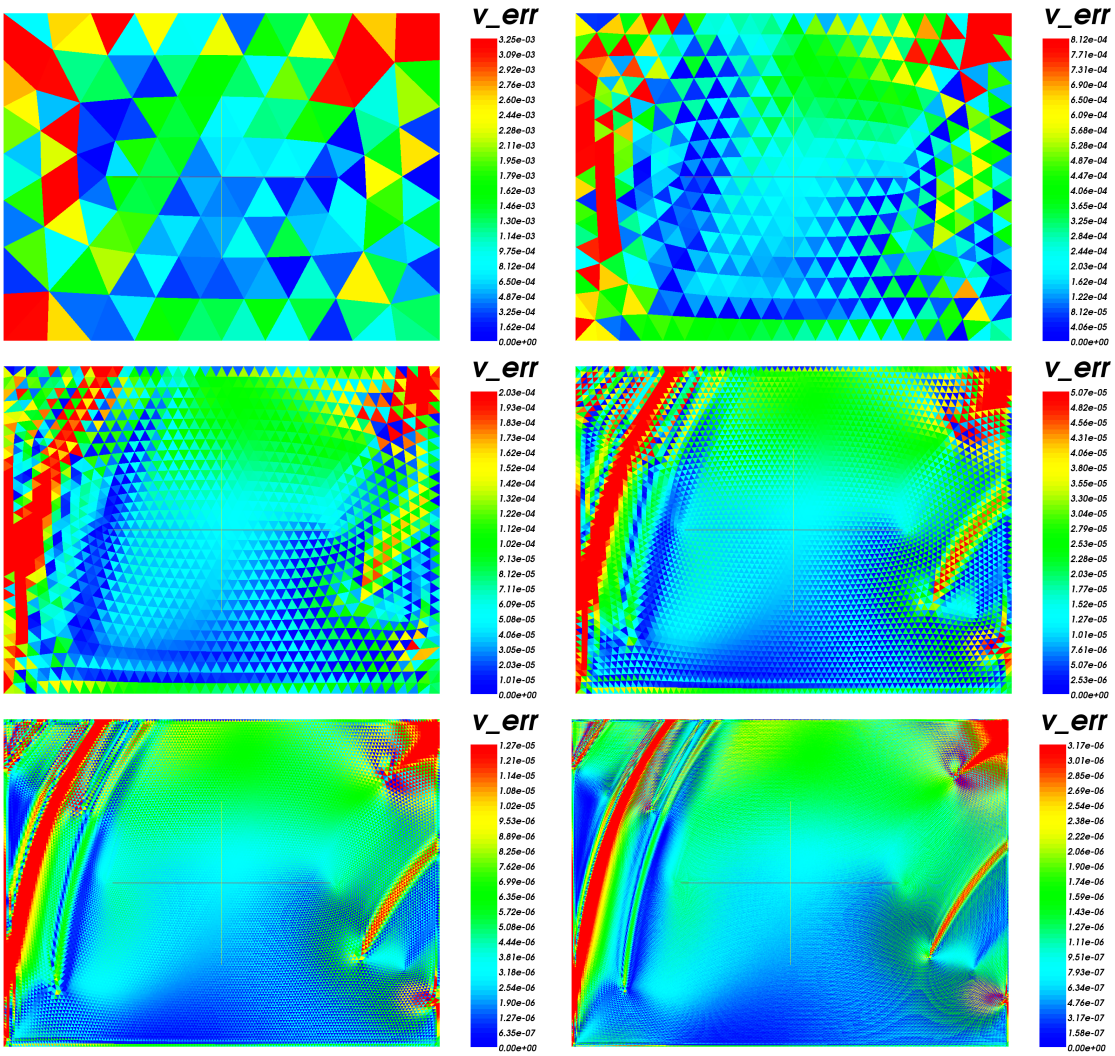


Figure B.27: Case 7: Set B, gradients with Gauss integration; y-velocity

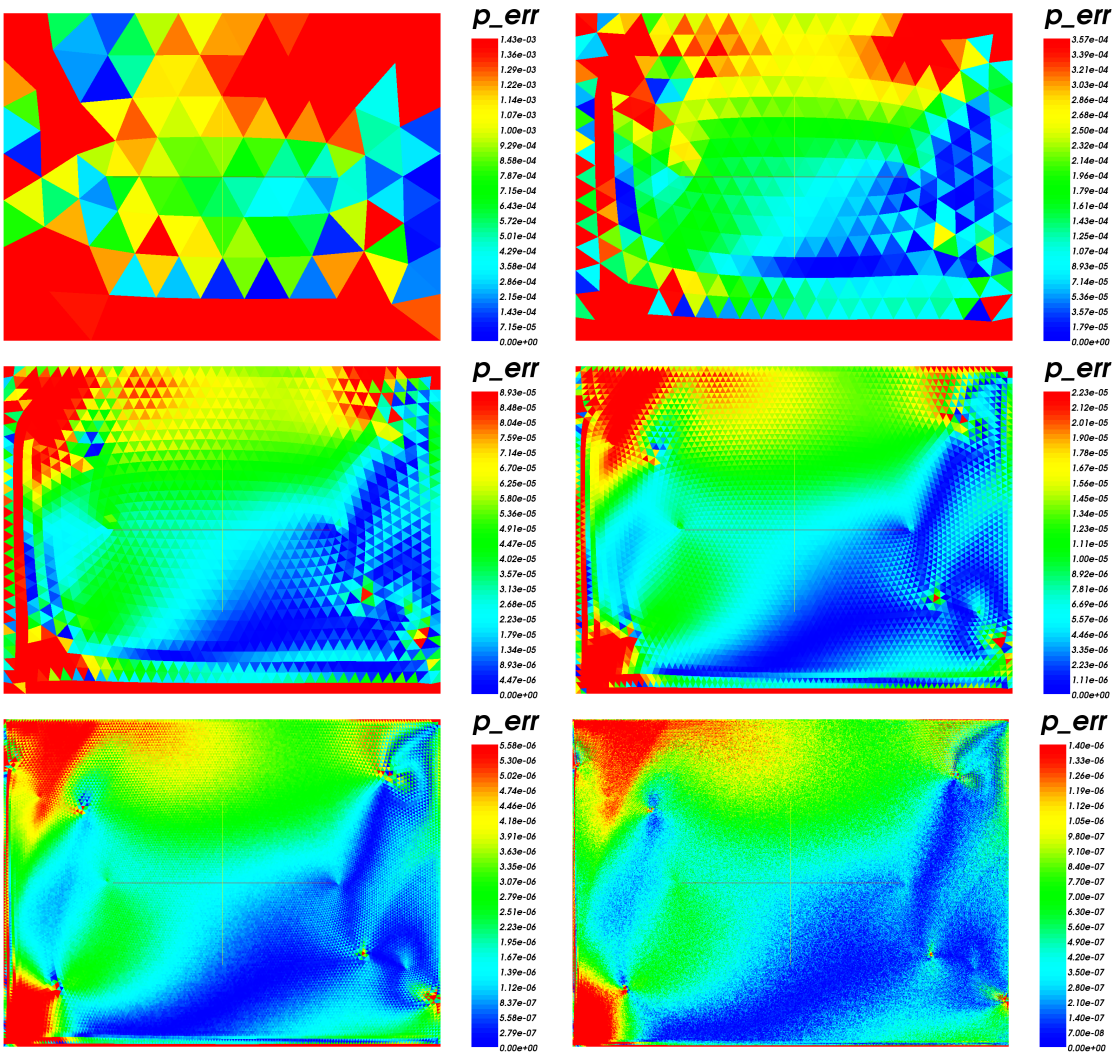
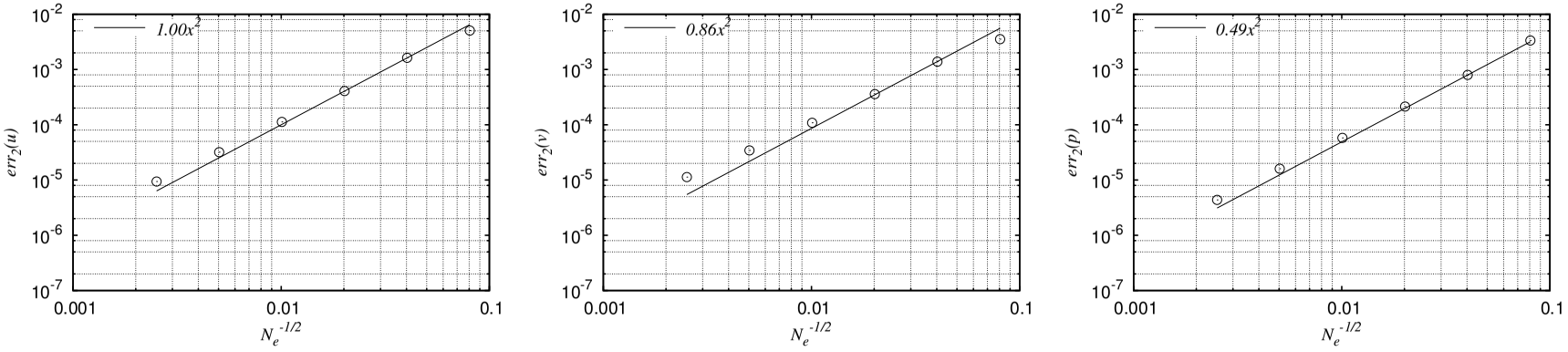


Figure B.28: Case 7: Set B, gradients with Gauss integration; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	9.38E-06	1.13E-05	4.37E-06	1.77	1.60	1.88	1.79	1.63	1.86	1.81	1.65	1.85
39424	3.19E-05	3.43E-05	1.60E-05	1.82	1.66	1.85	1.84	1.69	1.87	1.85	1.72	1.88
9856	1.12E-04	1.08E-04	5.78E-05	1.86	1.73	1.89	1.93	1.84	1.89	1.98	1.92	1.90
2464	4.08E-04	3.59E-04	2.14E-04	2.00	1.95	1.90	1.82	1.65	1.99	1.68	1.45	2.05
616	1.63E-03	1.39E-03	7.97E-04	1.64	1.35	2.07	-	-	-	-	-	-
154	5.06E-03	3.54E-03	3.35E-03	-	-	-	-	-	-	-	-	-

Figure B.29: Case 8: Set B, no skewness correction; order of accuracy

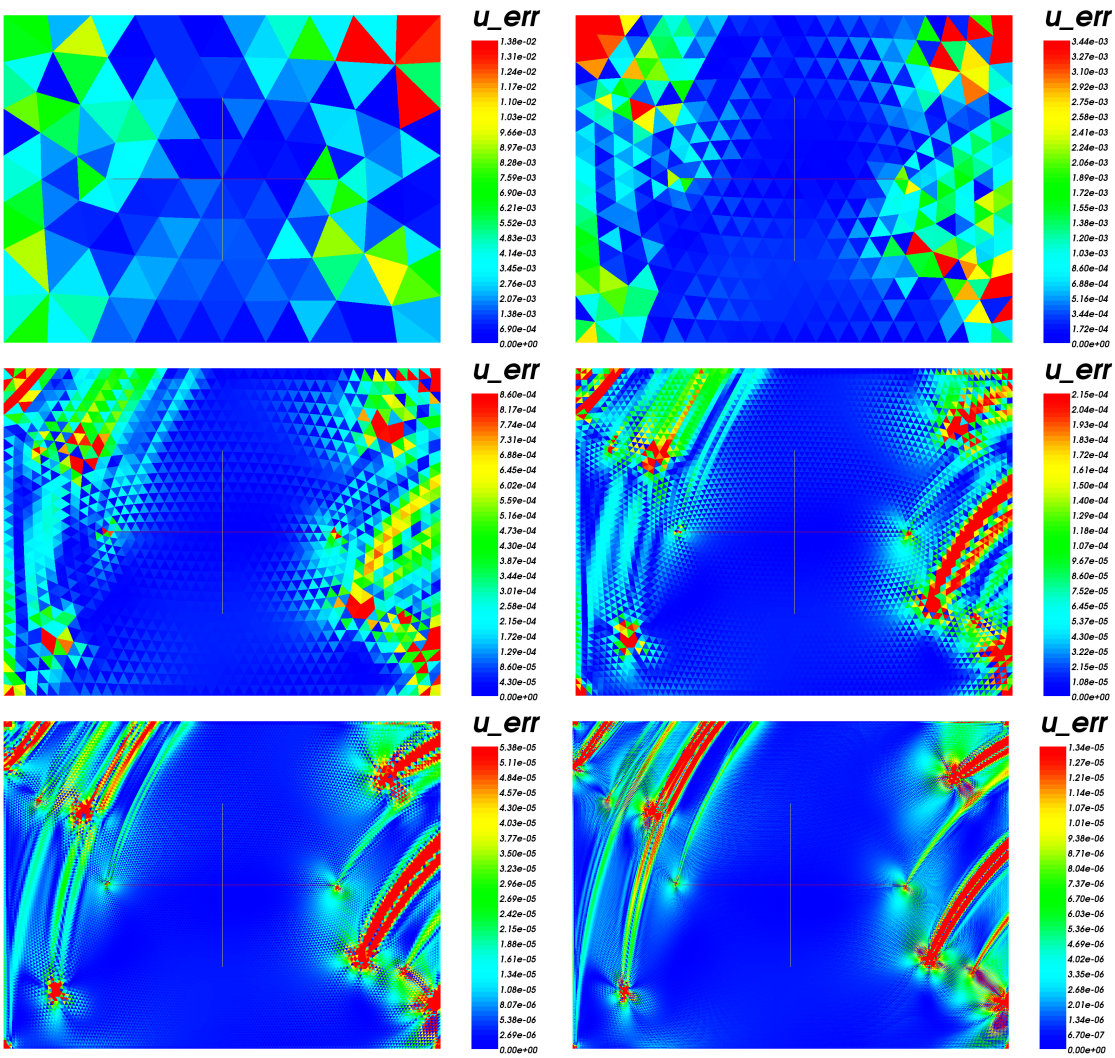


Figure B.30: Case 8: Set B, no skewness correction; x-velocity

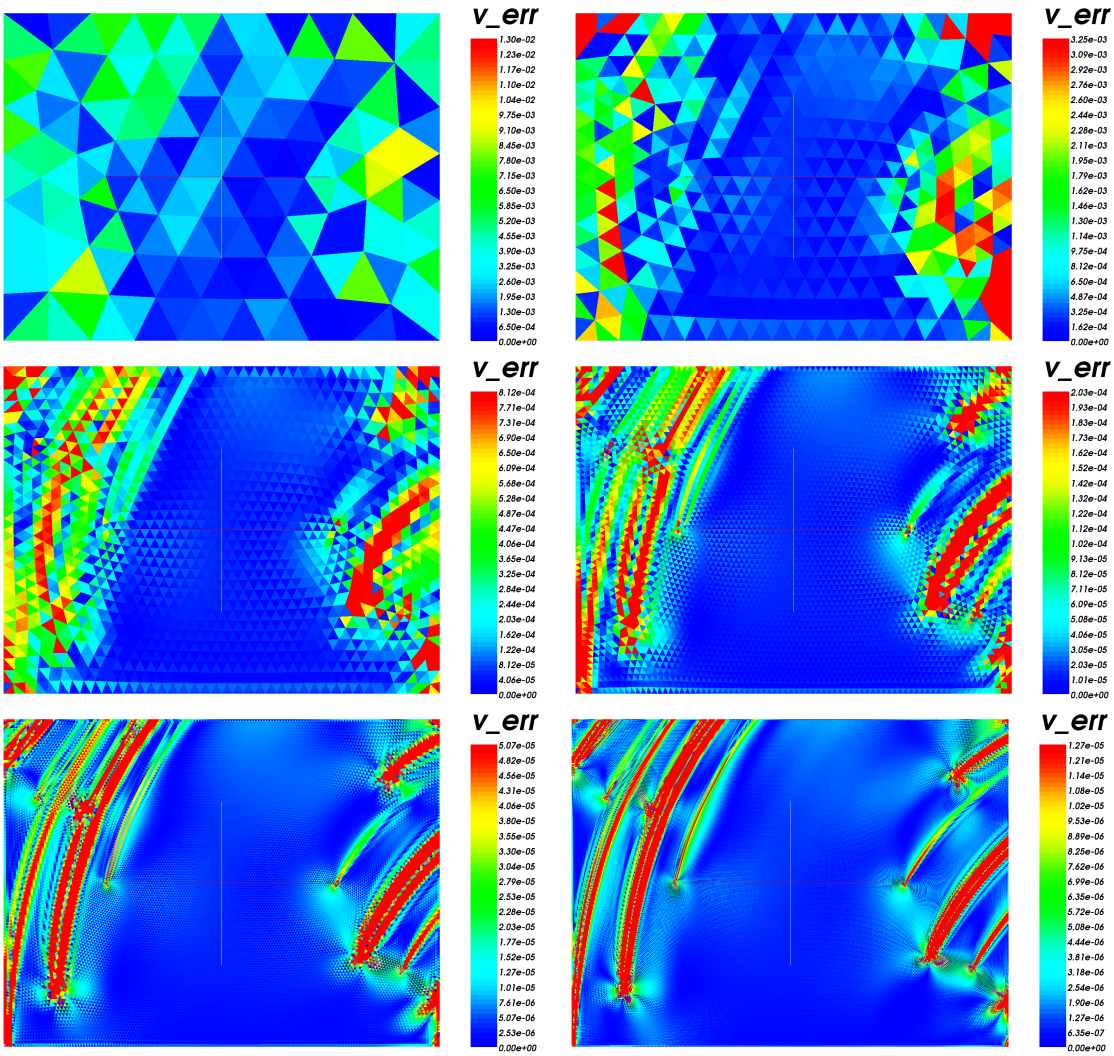


Figure B.31: Case 8: Set B, no skewness correction; y-velocity

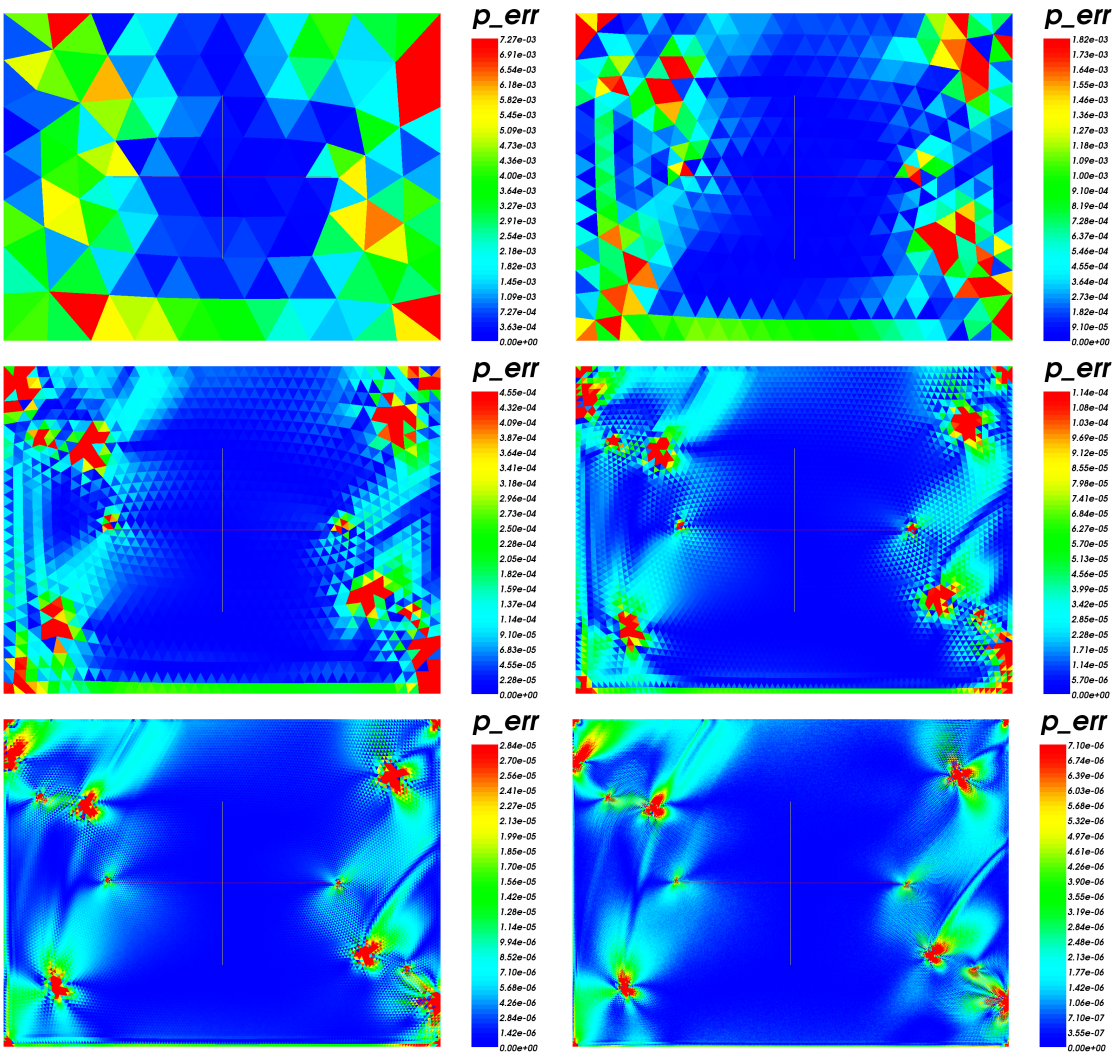
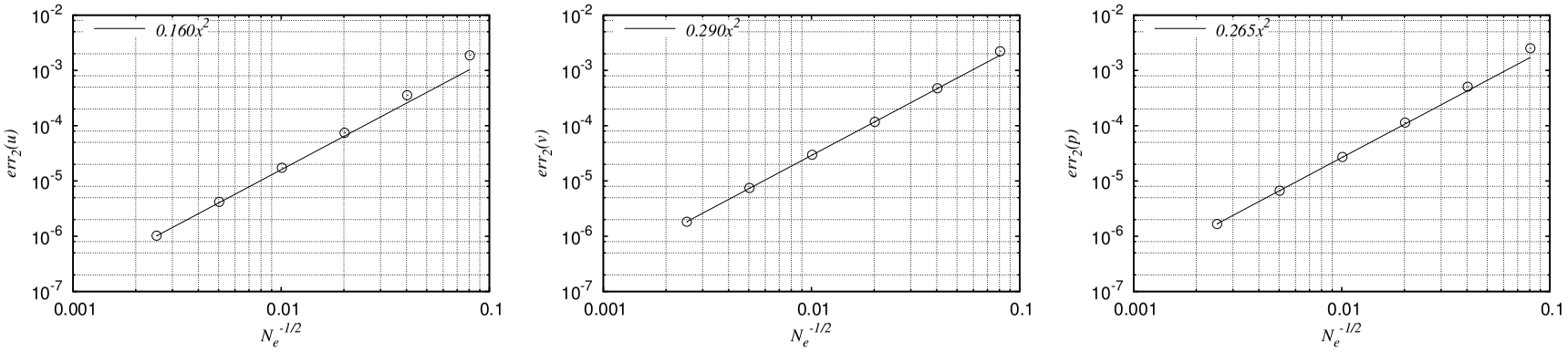


Figure B.32: Case 8: Set B, no skewness correction; pressure



N_e	$err_2(u)$	$err_2(v)$	$err_2(p)$	$p_u^{i,2}$	$p_v^{i,2}$	$p_p^{i,2}$	$p_{\log u}^{i,3}$	$p_{\log v}^{i,3}$	$p_{\log p}^{i,3}$	$p_u^{i,3}$	$p_v^{i,3}$	$p_p^{i,3}$
157696	1.02E-06	1.83E-06	1.66E-06	2.03	2.03	2.00	2.05	2.00	2.02	2.04	2.01	2.01
39424	4.18E-06	7.47E-06	6.66E-06	2.05	2.00	2.03	2.10	1.98	2.06	2.08	1.99	2.05
9856	1.73E-05	2.98E-05	2.71E-05	2.11	1.98	2.07	2.25	2.02	2.15	2.18	2.00	2.11
2464	7.45E-05	1.17E-04	1.14E-04	2.26	2.02	2.15	2.38	2.21	2.30	2.33	2.12	2.23
616	3.57E-04	4.75E-04	5.07E-04	2.39	2.22	2.32	-	-	-	-	-	-
154	1.87E-03	2.22E-03	2.52E-03	-	-	-	-	-	-	-	-	-

Figure B.33: Case 9: Set B, opposite node boundary condition; order of accuracy

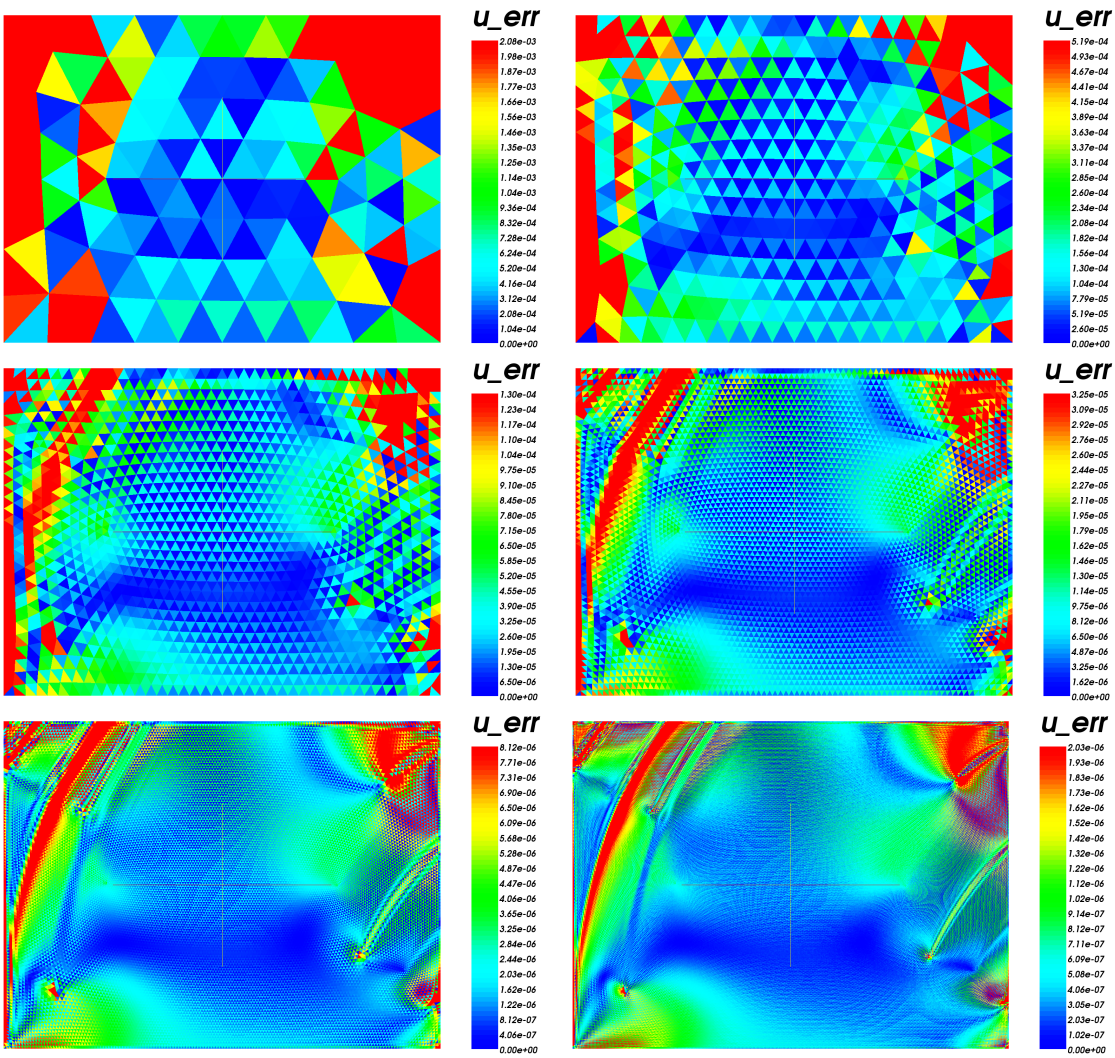


Figure B.34: Case 9: Set B, opposite node boundary condition; x-velocity

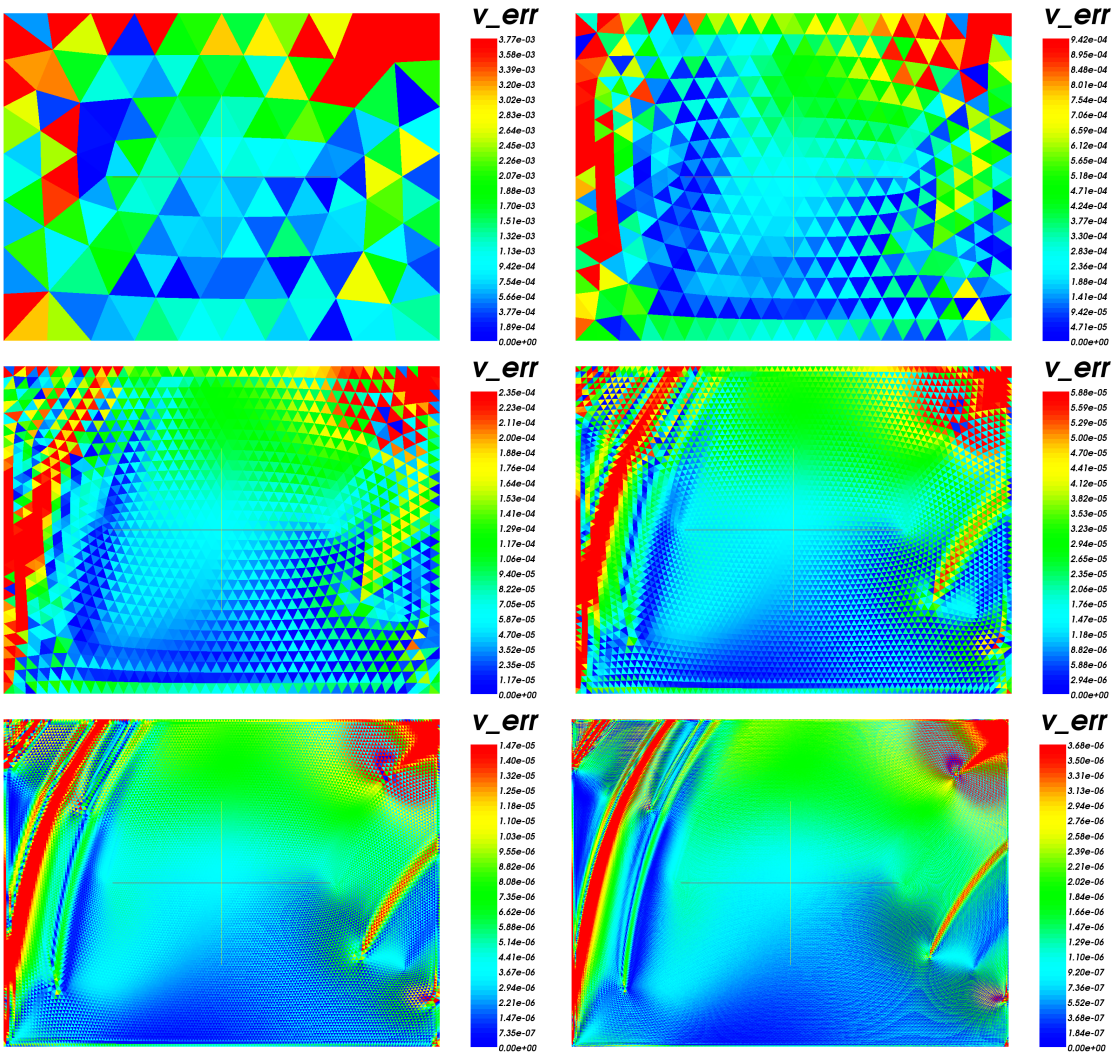


Figure B.35: Case 9: Set B, opposite node boundary condition; y-velocity

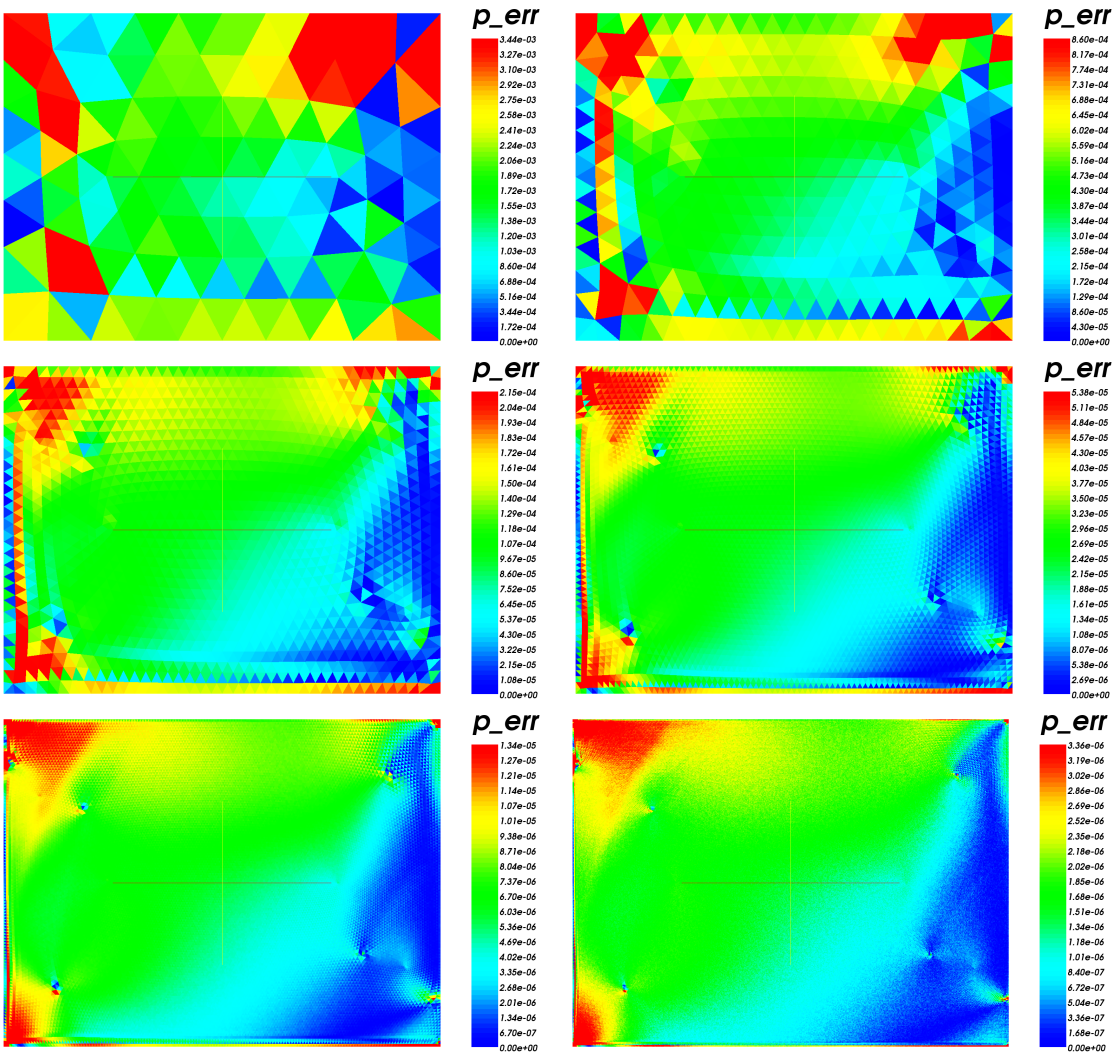


Figure B.36: Case 9: Set B, opposite node boundary condition; pressure

Appendix C The complete results for the free surface manufactured solution

This appendix presents the full results separately for each discretisation parameter combination in the standing wave manufactured solution case, which was discussed in Chapter 6. For each case the spatial and temporal distribution of the error in wave height are presented as colour maps. The colour ranges are scaled assuming a second-order accuracy for the method. First, a surface with the form

$$f(x, y) = ax^2 + by^2 \tag{C.1}$$

was fitted to the L_2-norms of the error presented in Fig. 6.2 using the non-linear fit option in gnuplot. Then the absolute value of the minimum and maximum of each colour range was taken as twice the value of the function (C.1) for the corresponding values of the discretisation parameters. The maps are grouped on the basis of the grid used, and for each grid the results with different time steps (coarsest first) are presented.

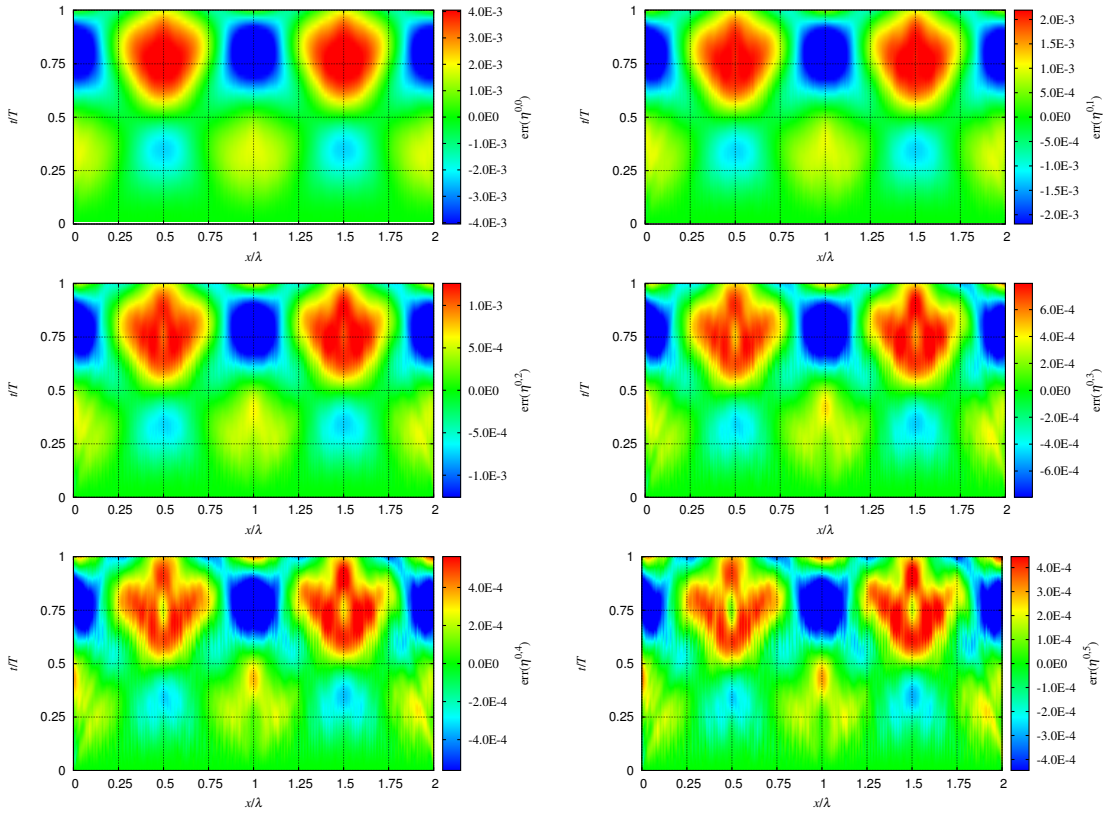


Figure C.1: Grid level 0: The error in wave height. The time step decreases from left to right and from top to bottom.

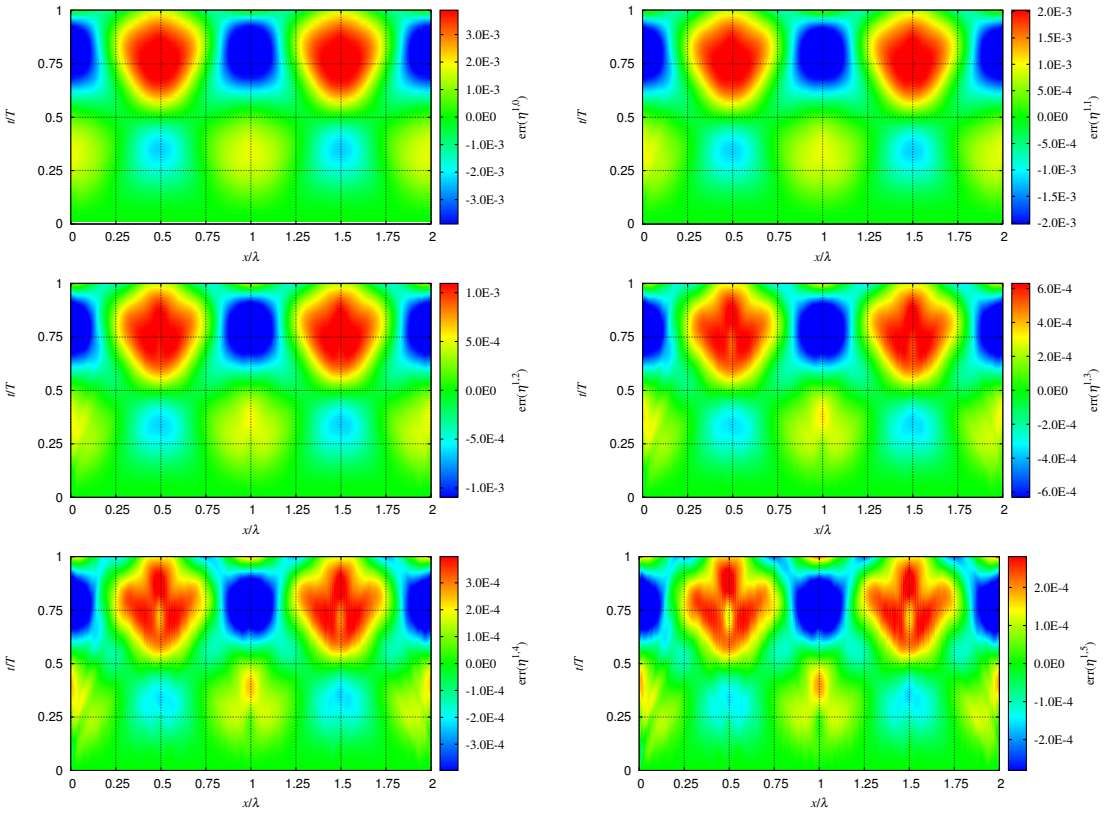


Figure C.2: Grid level 1: The error in wave height. The time step decreases from left to right and from top to bottom.

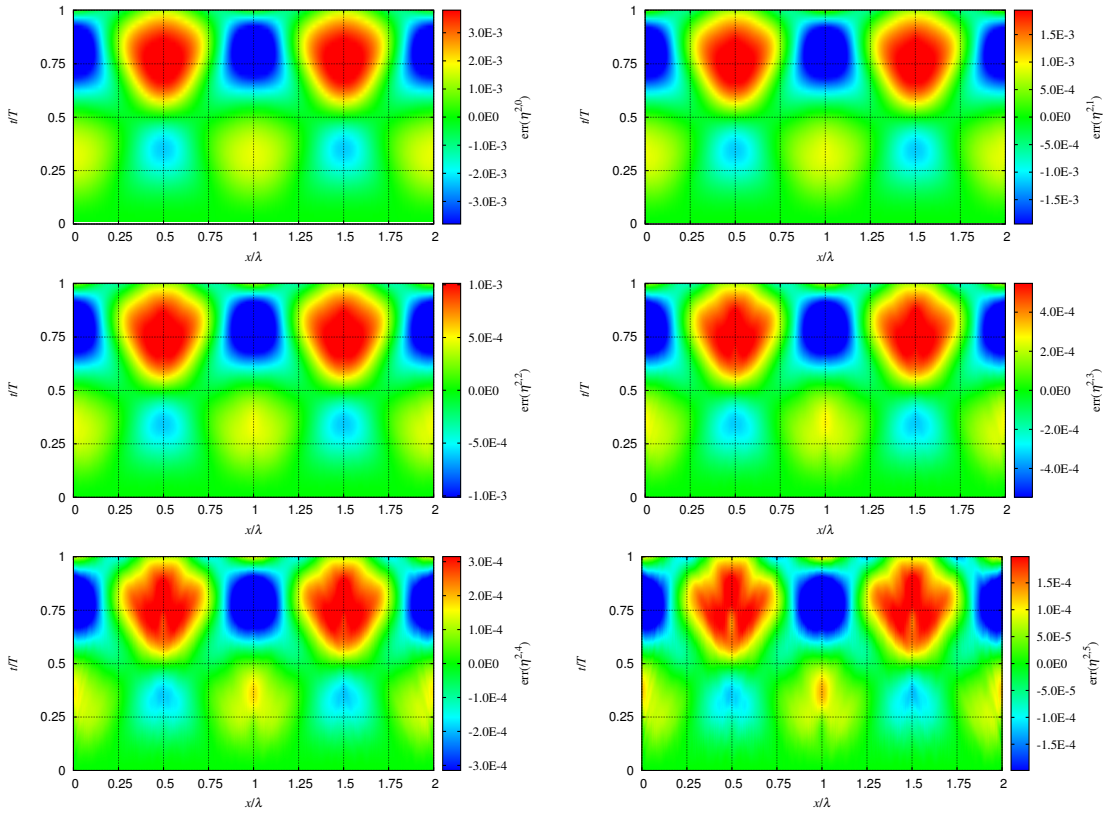


Figure C.3: Grid level 2: The error in wave height. The time step decreases from left to right and from top to bottom.

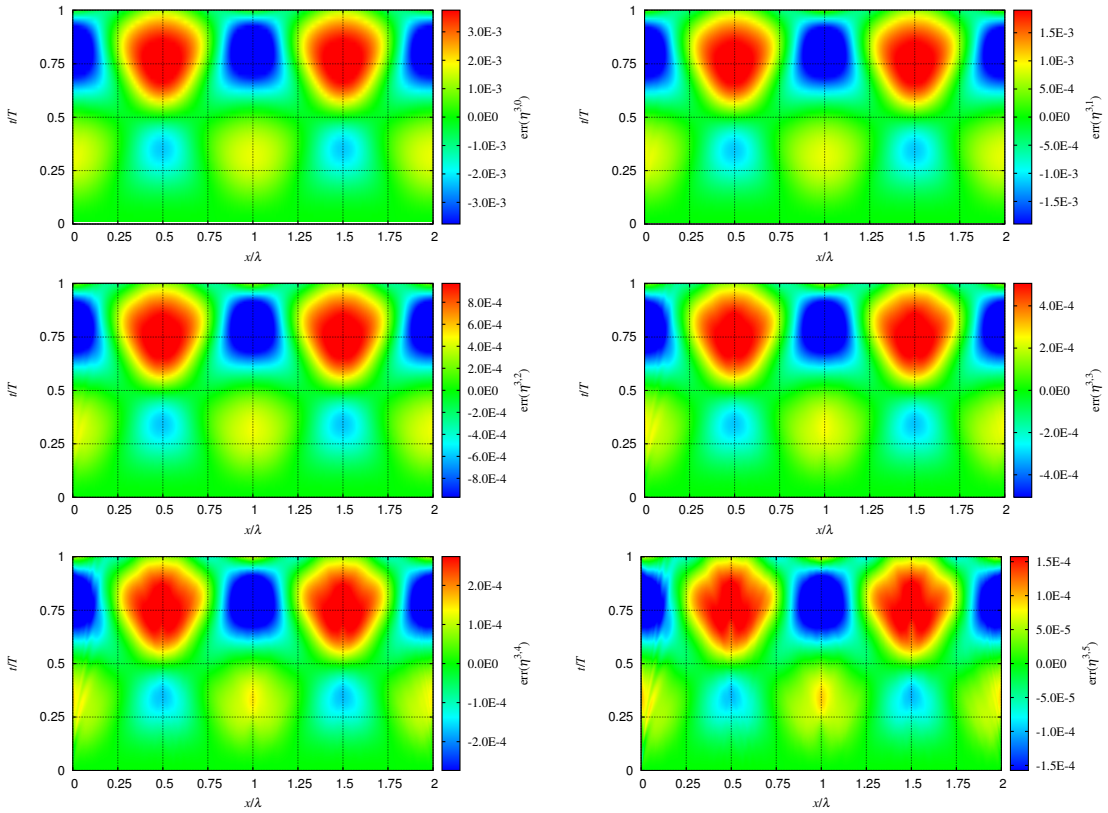


Figure C.4: Grid level 3: The error in wave height. The time step decreases from left to right and from top to bottom.

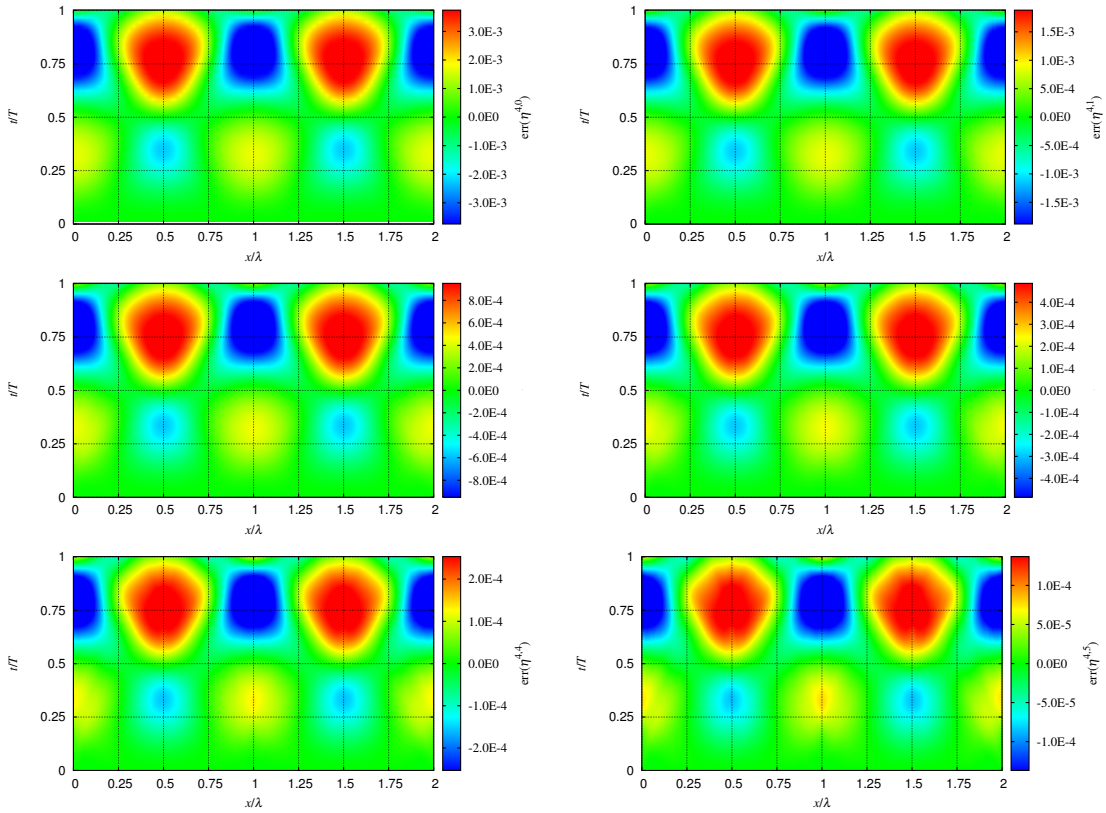


Figure C.5: Grid level 4: The error in wave height. The time step decreases from left to right and from top to bottom.

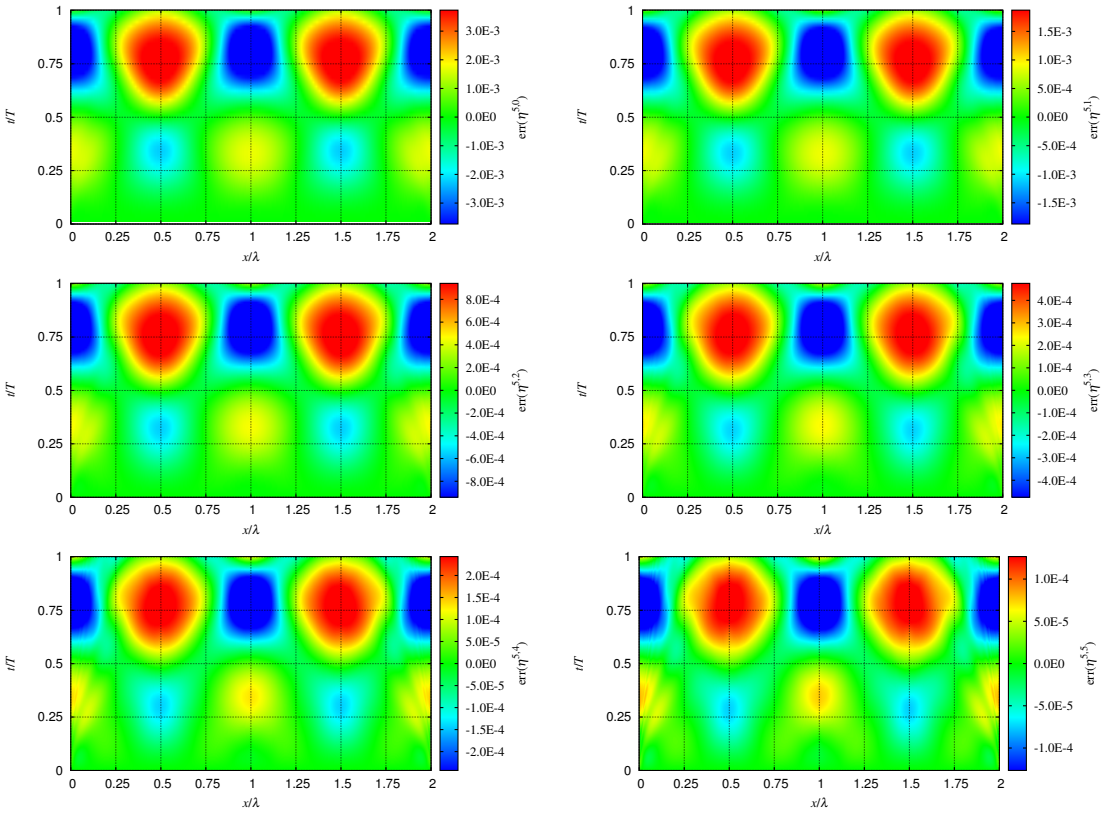


Figure C.6: Grid level 5: The error in wave height. The time step decreases from left to right and from top to bottom.

ISBN 978-952-248-136-8
ISBN 978-952-248-137-5 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)